IJCAI 2016 workshop

Natural Language Processing meets Journalism

Proceedings of the Workshop

Larry Birnbaum, Octavian Popescu and Carlo Strapparava (eds.)

10 July 2016
New York City

Preface

With the advent of the digital era, journalism faces what seems to be a major change in its history - data processing. While much journalistic effort has been (and still is) dedicated to information gathering, now a great deal of information is readily available but is dispersed in a large quantity of data. Thus processing a continuous and very large flow of data has become a central challenge in today's journalism.

With the recognition of this challenge, it has become widely accepted that data-driven journalism is the future. Tools which perform big data mining in order to pick out and link together what is interesting from various multi media resources are needed; these tools will be used as commonly as typewriters once were. Their scope is well beyond data classification. They need to construct sense and structure out of the never-ending flow of reported facts, ascertaining what is important and relevant. They need to be able to detect what is behind the text, what authors' intentions are, what opinions are expressed and how, whose propagandistic goal an article might serve, etc. What's more, they need to go beyond an intelligent search engine: They need to be picky and savvy, just like good journalists, in order to help people see what is really going on.

This volume contains the 18 papers presented at NLPMJ-2016: Natural Language Processing Meets Journalism, Workshop IJCAI-2016 held on July 10th, 2016 in New York. The workshop was at its first edition, and it attracted an unexpected number of submissions. In the call for papers, we welcomed reports on the recent progress on overcoming the bottlenecks in open domain relation extraction, paraphrasing, textual entailments and semantic similarity, and on their results in analyzing news content. However, we were also greatly interested in technologies for enhancing the communicative function of language in this context more generally, including NLP for automatizing creativity in advertising, or plagiarism for example.

We are happy to notice that the papers accepted to this workshop encompass a large subset of the topics we proposed for this workshop and quite a few of them focused on general phenomena occurring in mass-media that would be hard to substantiate without the use of NLP methods. Very interesting papers show that NLP is able to detect language gender biased, strong and deep resistance to external political factors which arbitrary make decision on the form of a language, both overt and hidden attitude , or polarity frames, in mass media. Also, the research focused on the denotation is well represented at this workshop, and some authors looked into satire, agreement, trends etc. We gladly witness the apparition of the new generation of tools which may be used by journalists and exploring topics, analyzing trends and making predictions, finding catching stories and head-lines, are among the things for which one can expect to have computers as assistants. The workshop also benefits from the direct interactions between journalists and NLP researchers and we want to thanks our special guests for their participation. We would like to thank the program committee members that worked hard to select the best works, and the IJCAI workshop organizers for their support during the whole process of planning and logistics.

June 30, 2016
New York

Larry Birnbaum
Octavian Popescu
Carlo Strapparava

Organizers

Lawrence Birnbaum	Northwestern University
Octavian Popescu	IBM T.J. Watson Research Center
Carlo Strapparava	FBK-irst

Program Committee

Enrique Alfonseca	Google
Lawrence Birnbaum	Northwestern University
Dan Cristea	"Alexandru Ioan Cuza" University
Song Feng	IBM
Radu Gheorghiu	Institutul de Prospectiva
Daniela Gifu	"Alexandru Ioan Cuza" University
Jay Hamilton	Stanford University
Mark Hansen	Brown Institute, Columbia School of Journalism
Orin Hargraves	University of Colorado
Daisuke Kawahara	Kyoto University
Zornitsa Kozareva	Yahoo! Labs
Rada Mihalcea	University of Michigan
Preslav Nakov	Qatar Computing Research Institute, HBKU
Vivi Nastase	Universität Heidelberg
Gözde Özbal	FBK-irst
Daniele Pighin	Google
Octavian Popescu	IBM T.J. Watson Research Center
Mattia Rigotti	IBM T.J. Watson Research Center
Paolo Rosso	Universitat de Valencia
Malmasi Shervin	Harvard Medical School
Carlo Strapparava	FBK-irst
Olga Uryupina	University of Trento
Marcos Zampieri	Saarland University
Torsten Zesch	Sprachtechnologie, University of Duisburg-Essen

Table of Contents

Tie-breaker: Using language models to quantify gender bias in sports journalism

Liye Fu and **Cristian Danescu-Niculescu-Mizil** and **Lillian Lee**
Cornell University
liye@cs.cornell.edu cristian@cs.cornell.edu llee@cs.cornell.edu

Abstract

Gender bias is an increasingly important issue in sports journalism. In this work, we propose a language-model-based approach to quantify differences in questions posed to female vs. male athletes, and apply it to tennis post-match interviews. We find that journalists ask male players questions that are generally more focused on the game when compared with the questions they ask their female counterparts. We also provide a fine-grained analysis of the extent to which the salience of this bias depends on various factors, such as question type, game outcome or player rank.

1 Introduction

There has been an increasing level of attention to and discussion of gender bias in sports, ranging from differences in pay and prize money[1] to different levels of focus on off-court topics in interviews by journalists. With respect to the latter, Cover the Athlete,[2] an initiative that urges the media to focus on sport performance, suggests that female athletes tend to get more "sexist commentary" and "inappropriate interview questions" than males do; the organization put out an attention-getting video in 2015 purportedly showing male athletes' awkward reactions to receiving questions like those asked of female athletes. However, it is not universally acknowledged that female athletes attract more attention for off-court activities. For instance, a manual analysis by Kian *et al.* [2009] of online articles revealed significantly more descriptors associated with the physical appearance and personal lives of male basketball players in comparison to female ones.

Transcripts of pre- or post-game press conferences offer an opportunity to determine quantitatively and in a data-driven manner how different are the questions which journalists pose to male players from those they pose to female players. Here are examples of a game-related and a non-game-relevant question, respectively, drawn from actual tennis interviews:

1. What happened in that fifth set, the first three games?
2. After practice, can you put tennis a little bit behind you and have dinner, shopping, have a little bit of fun?

To quantify gender discrepancies in questions, we propose a statistical language-model-based approach to measure how game-related questions are. In order to make such an approach effective, we restrict our attention in this study to a single sport—tennis—so that mere variations in the lingo of different sports do not introduce extra noise in our language models. Tennis is also useful for our investigation because, as Kian and Clavio [2011] noted, it "marks the only professional sports where male and female athletes generally receive similar amounts of overall broadcast media coverage during the major tournaments."

Using our methodology, we are able to quantify gender bias with respect to how game-related interview questions are. We also provide a more fine-grained analysis of how gender differences in journalistic questioning are displayed under various scenarios. To help with further analysis of interview questions and answers, we introduce a dataset of tennis post-match interview transcripts along with corresponding match information.[3]

2 Related Work

In contrast with our work, prior investigations of bias in sport journalism rely on manual coding or are based on simple lists of manually defined keywords. These focus on bias with respect to race, nationality, and gender [Rainville and McCormick, 1977; Sabo *et al.*, 1996; Eastman and Billings, 2001; Bruce, 2004; Billings, 2008; Kian and Clavio, 2011; Ličen and Billings, 2013]; see Van Sterkenburg *et al.* [2010] for a review.

Much of the work on gender bias in sports reporting has focused on "air-time" [Eastman and Billings, 2000; Higgs *et al.*, 2003]. Other studies looked at stereotypical descriptions and framing [Messner *et al.*, 1993; Jones, 2004; Angelini and Billings, 2010; Kian *et al.*, 2009]. For surveys, see Knight and Giuliano [2001] or Kaskan and Ho [2014], inter alia. Several studies have focused on the particular case of gender-correlated differences in tennis coverage [Hilliard,

[1] "U.S. Women, Fighting for Equal Pay, Win Easily as Fans Show Support", *The New York Times*, April 6, 2016. http://www.nytimes.com/2016/04/07/sports/soccer/uswnt-colombia-friendly-equal-pay-complaint.html

[2] http://covertheathlete.com/

[3] Dataset available at http://www.cs.cornell.edu/~liye/tennis.html

1984; Vincent *et al.*, 2007; Kian and Clavio, 2011]. We extend this line of work by proposing an automatic way to quantify gender bias in sport journalism.

3 Dataset Description

We collect tennis press-conference transcripts from ASAP Sport's website (http://www.asapsports.com/), whose tennis collection dates back to 1992 and is still updated for current tournaments. For our study, we take post-game interviews for tennis singles matches played between Jan, 2000 to Oct 18, 2015. We also obtain easily-extractable match information from a dataset provided by Tennis-Data,[4] which covers the majority of the matches played on the men's side from 2000-2015 and on the women's side from 2007-2015.

We match interview transcripts with game statistics by date and player name, keeping only the question and answer pairs from games where the statistics are successfully merged. This gives us a dataset consisting of 6467 interview transcripts and a total of 81906 question snippets[5] posed to 167 female players and 191 male players.

To model *tennis-game*-specific language, we use live text play-by-play commentaries collected from the website Sports Mole (http://www.sportsmole.co.uk/). These tend to be short, averaging around 40 words. Here is a sample, taken from the Federer-Murray match at the 2015 U.S. Open:[6]

> "The serve-and-volley is being used frequently by Federer and it's enabling him to take control behind his own serve. Three game points are earned before an ace down the middle seal [sic] the love hold."

For our analysis, we create a gender-balanced set of commentaries consisting of descriptions for 1981 games played for each gender.

4 Method

As a preliminary step, we apply a word-level analysis to understand if there appear to be differences in word usage when journalists interview male players compared to female players. We then introduce our method for quantifying the degree to which a question is game-related, which we will use to explore gender differences.

4.1 Preliminary Analysis

To compare word usage in questions, we consider, for each word w, the percentage of players who have ever been asked a question containing w. We then consider words with the greatest difference in percentage between male and female players.[7] The top distinguishing words, which are listed below in descending order of percentage difference, seem to suggest that questions journalists pose to male players are more game-related:

Male players: clay, challenger(s), tie, sets, practiced, tiebreaker, maybe, see, impression, serve, history, volley, chance, height, support, shots, server(s), greatest, way, tiebreaks, tiebreakers, era, lucky, luck;[8]

Female players: yet, new, nervous, improve, seed, friends, nerves, mom, every, matter, become, meet, winning, type, won, draw, found, champion, stop, fight, wind, though, father, thing, love.

4.2 Game Language Model

To quantify how game-related a question is in a data-driven fashion, we train a bigram language model using KenLM[9] [Heafield *et al.*, 2013] on the gender-balanced set of live-text play-by-play commentaries introduced in Section 3.

For an individual question q, we measure its *perplexity* $PP(q)$ with respect to this *game language model* $P_{\text{commentary}}$ as an indication of how game-related the question is: the higher the perplexity value, the less game-related the question. Perplexity, a standard measure of language-model fit [Jelinek *et al.*, 1977], is defined as follows for an N-word sequence $w_1 w_2 \dots w_N$:

$$PP(w_1 w_2 ... w_N) = \sqrt[N]{\frac{1}{P_{\text{commentary}}(w_1 \cdots w_N)}} .$$

Below are some sample questions of low-perplexity and high-perplexity values:

Perplexity	Sample Questions
Low	What about your serve, Rafa?
	The tiebreak, was that the key to the match?
High	Who designed your clothes today?
	Do you normally watch horror films to relax?

5 Experiments

In this section we use the game language model to quantify gender-based bias in questions. We then compare the extent to which this difference depends of various factors, such as question type, game outcome, or player rank.

5.1 Main Result: Males vs. Females

We first compute perplexities for each individual question[10] and then group the question instances according to the inter-

[4] http://www.tennis-data.co.uk/

[5] Each snippet represents one turn from one journalist. Most question snippets contain at least one question, although some could be merely clarifications or comments. Note that reporter information (who asked which question) is not available in the transcript.

[6] http://www.sportsmole.co.uk/tennis/wimbledon/live-commentary/live-commentary-roger-federer-vs-andy-murray-as-it-happened_232822.html

[7] Words that are gender-specific (like 'her') are manually discarded.

[8] It is interesting, but beyond the scope of this paper, to speculate on reasons why "luck" and "lucky" skew so strongly male.

[9] KenLM (https://kheafield.com/code/kenlm/) estimates language models using modified Kneser-Ney smoothing without pruning.

[10] We identify individual questions simply by looking for '?'.

viewee's gender class. Throughout we use the Mann-Whitney U statistical significance test,[11] unless otherwise noted.

Comparing perplexity values between the two groups, we find that *the mean perplexity of questions posed to male players is significantly smaller (p-value <0.001) than that of questions posed to female players. This suggests that the questions male athletes receive are more game-related.*

However, the number of interviews each player participates in varies greatly, with highly interviewed players answering as many as thousands of questions while some lesser-known players have fewer than 10 interview questions in the dataset. Thus it is conceivable that the difference is simply explained by questions asked to a few prolific players. To test whether this is the case, or whether the observation is more general, we micro-average the perplexities by player: for each of the 167 male players and 143 females who have at least 10 questions in our dataset, we consider the average perplexities of the questions they receive. Comparing these micro-averages, we find that it is still the case that questions posed to male players are significantly closer to game language (p-value $<$ 0.05), *indicating that the observed gender difference is not simply explained by a few highly interviewed players.*

5.2 Relation to Other Factors

We further investigate how the level of gender bias is tied to different factors: how typical the question is (section 5.2.1), the ranking of the player (section 5.2.2), and whether the player won or lost the match (section 5.2.3). For all the following experiments, we use per-question perplexity for comparisons: per-player perplexity is not used due to limited sample size.

5.2.1 Typical vs. Atypical Questions

One might wonder whether the perplexity disparities we see in questions asked of female vs. male players are due to "off-the-wall" queries, rather than to those that are more typical in post-match interviews. We therefore use a data-driven approach to distinguish between *typical* and *atypical* questions.

For any given question, we consider how frequently its words appear in post-match press conferences in general. Specifically, we take the set of all questions as the set of documents, D. We compute the inverse document frequency for each word (after stemming) that has appeared in our dataset, excluding the set S consisting of stop words and a special token for entity names.[12] For a question q that contains the set of unique words $\{w_1, w_2, ..., w_N\} \notin S$, we compute its *atypicality* score $Sc(q)$ as:

$$Sc(\{w_1, w_2, ..., w_N\}) = \frac{1}{N}\sum_{i=1}^{N} \text{idf}(w_i, D)\,.$$

We use the overall mean atypicality score of the entire question dataset as the cutoff point: questions with scores above the overall mean are considered atypical and the rest are considered typical.[13] Below are some examples:

Category	Sample Questions
Typical	Have you played each other before? How do you feel playing here?
Atypical	What about your haircut? Are you a vodka drinker?

Figure 1 shows that a gender bias with respect to whether game-related language is used exists for both typical and atypical questions. However, additional analysis reveals that the difference in mean perplexity values between genders is highly statistically significantly larger for atypical questions, suggesting that gender bias is more salient among the more unusual queries.

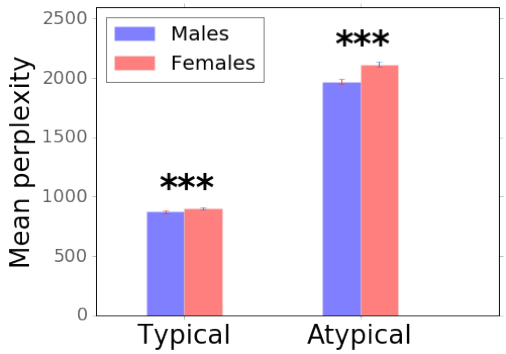


Figure 1: Mean perplexity values for male and female athletes after grouping the questions by how typical they are. Stars indicate high statistical significance ($p < 0.001$) between the male and female case. The male-female difference for the atypical group is statistically significantly larger than for the typical group.

5.2.2 Player Ranking

Higher ranked players generally attract more media attention, and therefore may be targeted differently by journalists. To understand the effect of player ranking, we divide players into two groups: top 10 players and the rest. For our analysis, we use the ranking of the player at the time the interview was conducted. (It is therefore possible that questions posed to the same player but at different times could fall into different ranking groups due to ranking fluctuations over time.) We find that questions to male players are significantly closer to game language regardless of player ranking (p-value < 0.001, Figure 2).

Furthermore, if we focus only on players who have ranked both in and outside the top 10 in our dataset, and pair the questions asked to them when they were higher-ranked to the questions asked when their ranking was lower, we find that there is no significant difference between questions asked to male athletes when they were in different ranking groups

[11] We used this non-parametric significance test instead of the t-test because it doesn't assume the samples to be normally distributed.

[12] We replace capitalized words and phrases with "<NOUN>"; for each word at the beginning of a sentence (which is always capitalized), we check whether it is a dictionary word.

[13] Questions consisting only of stop words and player or tournament names are still considered typical questions, even though they do not have an atypicality score.

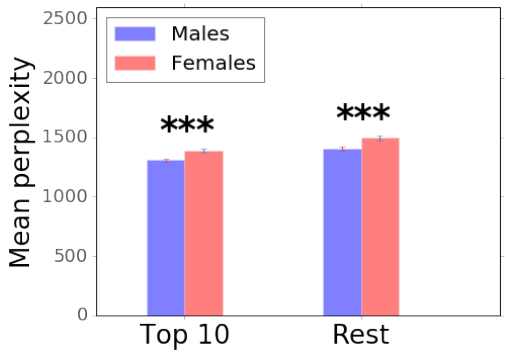


Figure 2: Mean perplexity values for male and female athletes after grouping the questions by the ranking of the player to which they are addressed. Stars indicate high statistical significance ($p < 0.001$) between the male and female case.

(Wilcoxon signed-rank p-value > 0.05). However, the difference is significant for females (Wilcoxon signed-rank p-value < 0.01), suggesting that gender bias may be more salient for lower ranked players as questions to lower-ranked female athletes tend to be less game-related.

While one might expect that star players would receive more off-court questions (yielding higher perplexities), the perplexity values for questions posed to top 10 players are actually lower regardless of gender. This may be because the training data for our language model is more focused on specific points played in matches, and may not be representative of tennis-related questions that are more general (e.g., longer-term career goals, personal records, injuries). In other words, our result suggests that journalists may attend more to the specifics of the games of higher ranked players, posing more specific questions about points played in the match during interviews.

5.2.3 Winning vs. Losing

While it is reasonable to expect that whether the interviewee won or lost would affect how game-related the questions are, the difference in mean perplexity for males and females conditioned on win/loss game outcome are comparable. In addition, for both male players and female players, there is no significant difference observed between the paired set of questions asked in winning interviews and the losing ones (Wilcoxon signed-rank p-value > 0.05), controlling for both player and season.[14] This suggests that that game result may not be a factor affecting how game-related the interview questions are.

6 Concluding discussion

In this work we propose a language-model based approach to quantify gender bias in the interview questions tennis players receive. We find that questions to male athletes are generally more game-related. The difference is more salient among the unusual questions in press conferences, and for lower-ranked players.

However, this preliminary study has a number of limitations. We have considered only a single sport. In addition, our dataset does not contain any information about who asked which question, which makes us unable to control for any idiosyncrasies of specific journalists. For example, it is conceivable that the disparities we observe are explained by differences in the journalists that are assigned to conduct the respective interviews.

In this work, we limit our scope to bias in terms of game-related language, not considering differences (or similarities) that may exist in other dimensions. Further studies may use a similar approach to quantify and explore differences in other dimensions, by using language models specifically trained to model other domains of interests, which may provide a more comprehensive view of how questions differ when targeting different groups.

Furthermore, our main focus is on questions asked during press conferences; we have not looked at the players' responses. The transcripts data, which we release publicly, may provide opportunities for further studies.

[14] We pair each question asked to a given player when winning to one question posed to the *same* player in the *same* calendar year when losing to construct the paired set of winning and losing questions for each gender.

Acknowledgments

We thank the anonymous reviewers and the participants in the Fall 2015 edition of the course "Natural Language Processing and Social Interaction" for helpful comments and discussion. This research was supported in part by a Discovery and Innovation Research Seed award from the Office of the Vice Provost for Research at Cornell.

References

[Angelini and Billings, 2010] James R. Angelini and Andrew C. Billings. An agenda that sets the frames: Gender, language, and NBC's americanized olympic telecast. *Journal of Language and Social Psychology*, 2010.

[Billings, 2008] Andrew C. Billings. *Olympic media: Inside the biggest show on television*. Routledge, 2008.

[Bruce, 2004] Toni Bruce. Marking the boundaries of the 'normal' in televised sports: The play-by-play of race. *Media, Culture & Society*, 26(6):861–879, 2004.

[Eastman and Billings, 2000] Susan Tyler Eastman and Andrew C. Billings. Sportscasting and sports reporting: The power of gender bias. *Journal of Sport & Social Issues*, 24(2):192–213, 2000.

[Eastman and Billings, 2001] Susan Tyler Eastman and Andrew C. Billings. Biased voices of sports: Racial and gender stereotyping in college basketball announcing. *Howard Journal of Communication*, 12(4):183–201, 2001.

[Heafield *et al.*, 2013] Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the ACL*, pages 690–696, August 2013.

[Higgs *et al.*, 2003] Catriona T. Higgs, Karen H. Weiller, and Scott B Martin. Gender bias in the 1996 olympic games:

A comparative analysis. *Journal of Sport & Social Issues*, 27(1):52–64, 2003.

[Hilliard, 1984] Dan C. Hilliard. Media images of male and female professional athletes: An interpretive analysis of magazine articles. *Sociology of Sport Journal*, 1:251–262, 1984.

[Jelinek *et al.*, 1977] Fred Jelinek, Robert L. Mercer, Lalit R. Bahl, and James K. Baker. Perplexity — a measure of the difficulty of speech recognition tasks. *The Journal of the Acoustical Society of America*, 62(S1):S63–S63, 1977.

[Jones, 2004] Dianne Jones. Half the story? Olympic women on ABC news online. *Media International Australia*, 110(1):132–146, 2004.

[Kaskan and Ho, 2014] Emily R. Kaskan and Ivy K. Ho. Microaggressions and female athletes. *Sex Roles*, 74(7-8):275–287, 11 2014.

[Kian and Clavio, 2011] Edward M. Kian and Galen Clavio. A comparison of online media and traditional newspaper coverage of the men's and women's U.S. open tennis tournaments. *Journal of Sports Media*, 2011.

[Kian *et al.*, 2009] Edward (Ted) M. Kian, Mondello Michael, and Vincent John. ESPN — the women's sports network? a content analysis of internet coverage of March madness. *Journal of Broadcasting & Electronic Media*, 53(3):477–495, 9 2009.

[Knight and Giuliano, 2001] Jennifer L. Knight and Traci A. Giuliano. He's a Laker; she's a "looker": The consequences of gender-stereotypical portrayals of male and female athletes by the print media. *Sex Roles*, 45(3-4):217–229, 2001.

[Ličen and Billings, 2013] Simon Ličen and Andrew C. Billings. Cheering for 'our' champs by watching 'sexy' female throwers: Representation of nationality and gender in Slovenian 2008 Summer Olympic television coverage. *European Journal of Communication*, 28(4):379–396, 2013.

[Messner *et al.*, 1993] Michael A. Messner, Margaret Carlisle Duncan, and Kerry Jensen. Separating the men from the girls: The gendered language of televised sports. *Gender & Society*, 7(1):121–137, 1993.

[Rainville and McCormick, 1977] Raymond E. Rainville and Edward McCormick. Extent of covert racial prejudice in pro football announcers' speech. *Journalism & Mass Communication Quarterly*, 54(1):20–26, 1977.

[Sabo *et al.*, 1996] Don Sabo, Sue Curry Jansen, Danny Tate, Margaret Carlisle Duncan, and Susan Leggett. Televising international sport: Race, ethnicity, and nationalistic bias. *Journal of Sport & Social Issues*, 20(1):7–21, 1996.

[Van Sterkenburg *et al.*, 2010] Jacco Van Sterkenburg, Annelies Knoppers, and Sonja De Leeuw. Race, ethnicity, and content analysis of the sports media: A critical reflection. *Media, Culture & Society*, 32(5):819–839, 2010.

[Vincent *et al.*, 2007] John Vincent, Paul M. Pedersen, Warren A. Whisenant, and Dwayne Massey. Analysing the print media coverage of professional tennis players: British newspaper narratives about female competitors in the Wimbledon championships. *International Journal of Sport Management and Marketing*, 2(3):281–300, 2007.

A Model for Multi-Perspective Opinion Inferences

Manfred Klenner
University of Zurich
Switzerland
klenner@cl.uzh.ch

Abstract

A text might give rise to various projections: a writer, a text and a reader projection. Given the (proclaimed) factuality of a text, the overt or hidden attitudes between the various referents can be inferred, as well as the writers opinion and - given the reader's preferences - his or her perception of the whole. Moreover, some sentences might even indicate controversial topics if viewed from a common sense perspective. We introduce an approach based on Description Logics that integrates these various perspectives into a joint model.

1 Introduction

Sentences might express a positive or negative relationship between people, organizations, nations etc. For instance, in the sentence "EU supports Greece" a positive attitude of the EU towards Greece is expressed. At the same time, a positive effect that is meant to be true is asserted. That is, Greece benefits from the situation described. If the reader has a positive attitude towards the beneficiary (Greece), he might regard the initiator (EU) as a benefactor and, thus, takes a positive attitude towards him as well (he is a proponent of his). If he does not like the beneficiary for some reasons, he might, as a consequence, regard the seemingly benefactor as his opponent. If the sentence is negated or embedded into a non-factive verb like "to pretend" ("EU pretends to support Greece") neither the positive relationship between the referents nor the positive effect on Greece do hold any longer. Instead, the matrix verb "to pretend" casts a negative effect on EU. If a positive effect in such a sentence is casted on an entity that from a common sense perspective is negative, then the actor of the described situation might be regarded as a common sense disturber (e.g. "The minister supports terrorism").

This is the kind of reasoning we have in mind. We would like to be able to answer the following questions: Given a text, what is good or bad *for* the entities mentioned in the text, what is good or bad *of* these entities, what are the attitudes of the entities towards each other and what follows from the reader's stance, i.e. his prior attitudes towards some entities, for his attitudes towards the entities mentioned in the sentence. The user of our system then could mine texts for proponents and opponents of his, in the sense that entities that do things (or like others that) he likes are proponents and entities that act in the opposite way (or like others he dislikes) are opponents. Also, controversial topics can be identified on the basis of a common sense perspective.

In contrast to existing work, we stress the point that verb signatures in the sense of [Karttunen, 2012] capturing (non-)factuality information regarding complement clauses need to be taken into account in order to properly draw such inferences. We focus on complex sentences where a matrix verb restricts its subclauses with respect to factuality depending on its affirmative status (i.e. whether the matrix clause is asserted or negated).

We have realized a joint model with Description Logics (DL), namely OWL [Horrocks and Patel-Schneider, 2011] and SWRL [Horrocks and Patel-Schneider, 2004]. The OWL model is language-independent, however, the parser and the lexicon resources are not. We rely on English examples, however our pipeline (and the empirical evaluation) is for German.

2 Related Work

An early rule-based approach to sentiment inference is [Neviarouskaya *et al.*, 2009]. Each verb instantiation is described from an internal and an external perspective. For example, "to admire a mafia leader" is classified as affective positive (the subject's attitude towards the direct object) given the internal perspective while it is (as a whole) negative externally. Factuality and subclause embedding do not play any role in their work. The same is true for [Reschke and Anand, 2011]. They capture the polarity of a verb frame instantiation as a function of the polarity of the verb's roles - we, instead, do not know in advance, but intend to infer the (contextual) polarity of the roles. Recently, [Deng and Wiebe, 2015] have introduced an advanced conceptual framework for inferring (sentiment) implicatures. Their work is most similar to our approach. Various model versions exist, the most recent one [Deng and Wiebe, 2015] also copes with event-level sentiment inference, which brings it even closer to our model. Probabilistic Soft Logic is used for the definition of the model and for drawing inferences. The goal of the systems is to detect pairs of entities that are in a PosPair or NegPair relation. However, factuality is not taken into account in their framework, while we believe it is crucial for certain inference steps.

How Description Logics can be used to identify so-called polarity clashes is described in [Klenner, 2015]. However, attitudes and the factuality of situations are not part of that model.

3 The Verb Model: Polarity Frames

The basis of our approach is a verb resource that we call polarity frames, cf. [Klenner and Amsler, 2016]. The current lexicon comprises 330 German verbs which gives 690 polarity frames. We are particularly interested in those verbs that subcategorize for complement clauses (78 verbs), since especially they are crucial for reasoning.

For each argument (agent, patient etc.) of a polarity frame we specify whether it casts a polar effect on its argument filler, e.g. the patient of "to help" gets a positive effect. We distinguish between effect roles that indicate that something is good/bad *of* or *for* someone. The agent role is an *of-role* - it is good *of* A to help B. The patient, but depending on the verb also the theme or recipient roles are *for-role*s, it is good *for* B if A helps him.

Take the verb "to help". There are at least two polarity frames, the transitive use (A helps B) and the one with an embedded (infinitival) subclause (A helps to XCOMP). In the first frame, both argument fillers receive a positive effect (in an affirmative, factual use of the verb). The agent is a positive *of-role*, which we call the *pos-of* role (a sub role of *of-role*). The patient is, accordingly, a *pos-for* role. Both roles are generalizations of the traditional roles (agent, ..). They ease the development of general inference rules and they have a particular function in the reasoning process. We would like to be able to state that something is good or bad *of or for* someone.

In the second frame (help to XCOMP), the agent again is the bearer of the *pos-of* role. But now it is the subclause that receives a positive effect, i.e. it is good for the situation denoted by the subclause that it receives help. Thus, not only entities but also situations are affected by the polarity a verb casts on its arguments. In order to distinguish roles for situations from roles for entities, we call the role for positively and negatively affected situations *poseff* and *negeff*, respectively.

3.1 Verb Signatures

Verbs that subcategorize for a clausal complement are further specified for (non-)factuality of the clausal complement. Factuality means that the situation described in the subclause is meant (by the writer) to be true (to hold). We follow the work of [Karttunen, 2012], who distinguishes factive, non-factive and implicative verbs. Factuality of the subclause depends on the (matrix) verb signature and the presence or absence of negation (in the matrix clause). Factive verbs such as "to regret" cast factuality on their subclause, whether the main clause is negated or not. If A regrets that COMP, then COMP is true in the sense that the speaker believes (or a least asserts) COMP to be true. The same holds for A does NOT regret that COMP. Sublcauses of non-factive verbs, on the other hand, are never meant to be true (e.g. "to pretend", "to hope").

Then there are verbs called implicatives that cast a mixture of factuality and non-factuality. Two-way implicatives like "to forget to" have non-factual subclauses in an affirmative use, but factual subclauses if negated. One-way implicatives only give rise to factuality in either the affirmative ("to force") or negated matrix verb contexts ("to refuse"). Table 1 summarizes the signatures, introduces the concept labels (e.g. clAF) we use to represent it and gives example verbs.

concept	explanation	matrix verb
clF	factual	to regret
clAF	factual, if affirmative	to force
clNaNF	non-factual, if non-affirmative	to manage
clANF	non-factual, if affirmative	to forget
clNaF	factual, if non-affirmative	to forget
clNaO	true or false, if non-affirmative	to help

Table 1: (Non-)Factuality of Subclauses

In Table 2 we give the polarity frames of some verbs.

		of	for	sc	aff	neg
1	to criticize	none	-	neg	clAF	clNaF
2	to refuse	none	-	neg	clANF	clNaO
5	to help	pos	pos	-		
7	to survive	-	pos	-		

Table 2: Polarity Frames

A hyphen indicates that the role is not part of the verb frame in question, pos and neg stand for positive and negative effect, respectively and $none$ states that although the argument role exists, there is no (i.e. a neutral) effect attached to it (sc means subclause effect). The last two columns relate to the verb signatures as introduced in Table 1, the forelast column reports the restriction if the matrix verb is aff(irmative) and the last column if it is neg(ated). For example, the sublcause of "to refuse" (row 2) is non-factual if the refuse sentence is affirmative (clANF), but its truth value is unspecified (clNaO) if negated.

4 Description Logics Model

We strive to be able to combine different perspectives in a joint model. Firstly, there is the question of who actually profits (or has a disadvantage) from the described situation. We call this the layer of *effect projection*. Then there is the relational level that determines the attitudes of the participants towards each other, this is called the *attitude projection*. Both are derived from the input text, they represent, so to speak, the way the text puts the world (the *text perspective*). There is also the perspective of the reader, the *reader projection* and the perspective of the author (not coped with in this paper). Finally, we also deal with what we call the common sense perspective. Here we focus on the detection of controversial topics where a polarity conflict occurs given the sentence.

Inferences are based on the text perspective, i.e. the view of the world that the author of the sentence intends to establish with his text. From the *text perspective* the attitudes of the author sometimes are evident, but in the kind of sentence that we envisage, this is normally not the case. We focus on sentences that report the view of the subject of the matrix clause ("A criticizes that ..").

Effect	Attitude	Reader
beneficiary	pro	MyOpponent
benefactor	contra	MyProponent
victim	cs_disturber	SympathyEntity
villain		NonSympathyEntity

Table 3: Projections: Concepts and Properties

Description Logics seemed to be well suited for such intermingled inference tasks. One must not care about the concrete sequence the inferences are drawn and there is the notion of global consistency that might help to identify and get rid of unwanted side effects. It turned out to be convenient to use SWRL rules [Horrocks and Patel-Schneider, 2004] instead of pure OWL concepts [Horrocks and Patel-Schneider, 2011] to define the relational inference layer. Our system was developed on the basis of the Protégé editor, HermiT [Glimm *et al.*, 2014] was used as a SWRL and OWL reasoner.

4.1 Overview: Concepts and Properties

Table 3 shows the concepts and properties of the various projection layers. We give a brief description of the overall system in order to instantiate the OWL constructs from Table 3. We use a dependency parse tree as input. A simple rule-based component (see [Klenner and Amsler, 2016]) extracts the grammatical roles (subject etc.) of each verb from the parse trees (thereby normalizing passive voice and making implicit arguments explicit, i.e. given control or raising verbs). The output of this component are the instantiated verb frames, i.e. the filler objects of the grammatical roles of the verbs given the sentence. Each grammatical role then is mapped to a polar role (*pos-of*, *neg-of* etc.). If we know the grammatical role of a referent then we know his polar role, that is the core functionality of our polarity frames. The next step is to produce the OWL representation of the sentence (see section 4.3). For every verb its affirmative status is given by the parse tree (this becomes also part of the OWL representation). Whether the main clause is factual or not is determined by a simple heuristics: if no modal verbs or modifiers are present, then the sentence is factual. Factuality of subclauses are predicted by OWL definitions.

In a nutshell this is how the various layers from Table 3) interact. Take "EU no longer supports Greece". Here, "Greece" is victim – it suffers from the situation. From the parse tree we know that it is the direct object of "support", from the polarity lexicon we known that the direct object is a *pos-for* role. Since the sentence negated, the *pos-for* gets inverted and becomes a *neg-for* role. Now the OWL definition of a victim is met (see section 4.4). This is an example of an effect projection. Furthermore since "EU" is responsible for a negative effect on "Greece", it must have a negative attitude towards "Greece", a *contra* relation if found (an attitude projection). Finally, if "Greece" is a *SympathyEntity* of the reader (the concept representing the reader's prior attitudes), then "EU" becomes an instance of *MyOpponent* of the reader. If the sentence was "EU supports neoliberal greed", then "EU" becomes a common sense disturber (cs_disturber) since a polarity conflict occurs. A positive effect on a negative denotation ("neoliberal greed") is found, which is from a common sense perspective not desirable, it represents a conflict, thus.

4.2 Properties

OWL properties represent two-placed relations between concepts, they have domain and range restrictions (we do not specify the concrete restrictions here). We have a property *for-role* with sub properties *pos-for* and *neg-for* and a property *of-role* with *pos-of*, *neg-of* as sub properties. These are roles for entities, for situations we use a general role *cl-role* denoting a non-polar subclause (e.g. the verb "to remember" (that) would have it) and *negeff* and *poseff* for positive and negative effects, the matrix verb casts on its complement clause. These roles also have inverse roles, indicated by an preceding initial I (e.g. *I-pos-of*). Table 4 summarizes these properties. They are use to represent an input sentence, i.e. the instantiated verb frames. We now turn to this part of model. Please note that, in contrast, the properties of the attitude projection (cf. Table 3, second column) are subject to SWRL inference rules (see section 5).

of-role	the agent
(pos|neg)-of	the filler gets a positive (negative) effect
for-role	the patient,recipient, beneficiary or theme
(pos|neg)-for	a positive (negative) for-role
cl-role	the subclause
(pos|neg)eff	subclause receives a positive (negative) effect

Table 4: Properties Representing Verb Argument Roles

4.3 A-Box Representation

We represent verb instantiations in a manner that is inspired by Davidson's approach [Davidson, 1967]. Our example sentence, "The minister has criticized that the EU has helped Greece to survive" is represented by the assertions from Table 5 (the specifications are given in a slightly simplified Manchester syntax, cf. [Horridge *et al.*, 2006]).

criticize-1 : (aff AND clAF)	help-1 : (aff AND clAF)
criticize-1 of-role minister-1	help-1 pos-of EU
criticize-1 negeff help-1	help-1 pos-for Greece
survive-1 : affirmative	help-1 poseff survive-1
survive-1 pos-for Greece	

Table 5: A-Box Representation

$criticize$-1 is a instance of both, the classes *aff*irmative and *clAF* (and, not shown here, *clNaF*), it has e.g. the role *negeff* with $help$-1 as its filler. The concepts *aff*irmative and *non-aff*irmative are used to represent the affirmative or negated use of a verb predicate in a sentence. The individuals $minister$-1, EU and $Greece$ are all instances of a general concept called *RealWorldEntity*.

4.4 T-Box

As mentioned, we distinguish between the perspective of the reader, *MyView*, and the perspective of the text, *TextView*, see Fig.1. *TextView* tells us, what the author believes to be true. One task of the reader as part of the understanding of a text is to find out what the text entails (class *Implication*)

about the described situation (class *Situation*). A situation is either affirmative (class *affirmative*) or negated (class *non-affirmative*), which is known given the sentence (thus, both are primitive concepts).

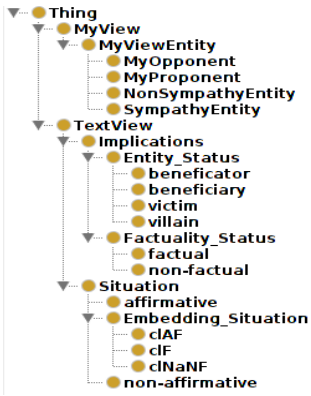


Figure 1: T-Box

The factuality of an embedd*ed* situation depends on the factuality class (e.g. clAF) of the embedd*ing* situation (class *Embedding_Situation*) given by the verb signature of the (embedding) verb (see Table 1 for the subclasses of *Embedding_Situation* not shown in Fig.1). For instance, according to Table 5 $criticize$-1 is an instance of *clAF* since the verb "to criticize" bears that signature: whatever affirmative $criticize$ embeds, it is factual[1]. Thus, all subclasses of *Embedding_Situation* are primitive concepts. Whether an embedded situation is factual or non-factual (its *Factuality_Status*) depends on the factuality class of the embedding verb and whether the embedding verb is affirmative or non-affirmative: *factual* and *non-factual* are defined classes. The definition of *factual* is (in Manchester syntax):

(I-cl-role some (clF or (affirmative and clAF) or (non-affirmative and clNaF)))

I-cl-role is the inverse of *cl-role*.

A situation is *factual* if it is embedded (*I-cl-role*) into a situation that is described by a factive verb (class *clF* from Table 1), or is *affirmative* and has the signature *clAF* or is *non-affirmative* and of type *clNaF*. Given this (together with the definition of *non-factual*), we are able to determine the factuality status of an embedded situation of any depth of embedding.

We now turn to the concept *Entity_Status*. We distinguish four classes and call them programmatically *benefactor*, *beneficiary*, *victim* and *villain*. We just give the definition of *beneficiary*. The idea behind our definition is that the beneficiary of a situation is somebody who benefits from it independently of any attitude that somebody might have towards him. So if A wins, A is the beneficiary, if A is liked by someone or not. What must be the case is that A occupies the *pos-for* role of a situation that is *factual* (not just imagined) and *affirmative*. Here is the definition of *beneficiary*:

(I-pos-for some (affirmative and factual))

[1]Clearly in: "A criticizes that B intends to lie", the intention is factual, not the lying.

5 SWRL Model: Attitude Projection

The main goal is to find out, whether A is for B, which we model with the property pro; or whether A is against B, here $contra$ is used.

Firstly, a verb might directly reveal the relation between the participants within the same clause: if A supports B, then A is pro B. Provided, of course, the situation is *factual*. In our SWRL rules the following class abbreviations are used: fact=*factual*,aff=*affirmative*,neg=*non-affirmative*,pfor=*pos-for*,nfor=*neg-for*.

```
r1 fact(?s),aff(?s),pfor(?s,?y),of_role(?s,?x)
     -> pro(?x,?y)
```

The first rule (variables are indicated by a leading question mark, e.g.?x) (r1) states: An actor ?x (the *of-role*) is pro ?y if in a factual, affirmative sentence ?s, ?y is the filler of the *pfor* role (e.g. "A supports B" gives pro(A,B)).

If a sentence ?s embeds a sentence ?s2, then rules like the following are in charge:

```
r5 aff(?s),fact(?s),aff(?s2),negeff(?s,?s2),
   of_role(?s,?x),nfor(?s2,?y)
      -> pro(?x, ?y)
r7 aff(?s),fact(?s),neg(?s2),negeff(?s,?s2),
   of_role(?s,?x),nfor(?s2,?y)
      -> contra(?x, ?y)
```

According to r5 an affirmative and factual clause ?s that embeds an affirmative subclause ?s2 bearing a negative effect (*negeff*) gives rise to a pro relation between the *of-role* of the matrix clause and the *nfor* role of the subclause. If A criticizes (clAF) or fears (clANF) that B punishes C, then A is pro C.

The agent-patient relation of rule r5 only holds if both, the matrix ?s and the subclause ?s2 are affirmative. If ?s2 is negated (cf. rule r7), then pro turns into $contra$ (A criticizes that B does not punish C gives contra(A,C)).

More complicated scenarios arise in the case of multiple embeddings. We discuss this given the two example sentences: 1) A criticizes that B refuses to help C and 2) A criticizes that B not refuses to help C. The task here is to fix the attitude of the subject of the matrix clause wrt. to any role at any level of subclause embedding. According to Table 2, both "to criticize" and "to refuse" put a negative effect on their complement clauses. We could say then that A (the matrix subject of example sentence 1) disapproves the negative effect of refuse, and thus *approves* the help situation (all this provided that the matrix situation is affirmative; the intermediate subclause must be affirmative as well; no information is needed wrt. to the innermost subclause).

```
r8 aff(?s),fact(?s),aff(?s2),of_role(?s,?x),
   negeff(?s,?s2),negeff(?s2,?s3)
     -> approve(?x,?s3)
r9 aff(?s),fact(?s),neg(?s2),of_role(?s,?x),
   negeff(?s,?s2),negeff(?s2,?s3)
     -> disapprove(?x,?s3)
```

```
r10 aff(?s),fact(?s),aff(?s2),of_role(?s,?x),
    negeff(?s,?s2),poseff(?s2,?s3)
     -> disapprove(?x,?s3)
```

That is: A *negeff* on a *negeff* gives (if ?s and ?s2 are affirmative) *approve* (see r8, sentence 1). If ?s2 is negated (?s is affirmative), a *negeff* on a *negeff* gives *disapprove* (see r9, sentence 2). The next rules describe how *approve* and *disapprove* propagate to pro or $contra$ properties.

```
r11  approve(?x,?s),aff(?s),pfor(?s,?y)
      -> pro(?x,?y)
r13 disapprove(?x,?s),aff(?s),pfor(?s,?y)
      -> contra(?x,?y)
```

If someone approves an affirmative situation that is positive (*pos-for*) for someone, then he is for this person (rule r11). Sentence 1: A is pro C (rule r8 and r11). According to rule r9 and r13, A is contra C (sentence 2).

Finally, some rules are used to propagate $contra$ and pro to derived $contra$ and pro properties. According to rule r15, if A is against B and B is against C then A is for C.

```
r15 contra(?x, ?y),contra(?y,?z)-> pro(?x,?z)
r18 pro(?x,?z),contra(?y,?z) -> contra(?x,?y)
```

Take: "A hopes that B does not offend C". Here, A is for C, but there is no inference regarding A's attitude towards B. However, if we know (e.g. from world knowledge) that B is against C, i.e. contra(B,C), then we can derive that A (presumably) is against B (rule r18: pro(A,C),contra(B,C) thus contra(A,B)).

6 Example

Take the following (hypothetical) sentence with the A-Box representation given in Table 5: "The minister has criticized that the EU has helped Greece to survive". The following inferences take place.

```
Beneficiary(Greece)          by OWL definition
pro(EU,Greece)                     by rule r1
contra(minister-1,EU)              by rule r8
disapprove(minister-1,survive-1)   by rule r10
contra(minister-1,Greece)          by rule r13
```

If Greece is an instance of *SympathyEntity*, it follows (by OWL definitions and the derived pro and $contra$) that

```
MyProponent(EU)
MyOpponent(minister-1)
```

7 Empirical Evaluation of the Core Model

We have implemented a prototype system for German: a verb frame extractor and converter to A-Box representations. There is no annotated German corpus available, so we created a gold standard of 50 sentences from newspaper texts. A sentence in order to get selected was required to have at least two verbs from our lexicon and two named-entities as role fillers of these verbs. From the first 1000 sentences we get from the WaCky corpus [Baroni M., 2009], we randomly selected 50 and annotated them for pro, $contra$, *beneficiary*,*victim*. A f-measure of 84.39% was achieved, the precision was 82.94%. and the recall was 85.88%. A error analysis revealed that parsing errors (subject mistaken as object etc.), missing polarity frames (especially prepositional phrases) and verb ambiguity are the main causes for the errors.

8 Common Sense Conflicts

In the core model, every actor who has according to the attitude projection a positive attitude (a pro relation) towards an instance of the reader's *NonSympathyEntity* is an *Opponent* of the reader. The class *NonSympathyEntity* is meant to capture those real-word entities the reader does not like - particular political parties, politicians, etc. These are personal preferences. But what about entities whose polar value is culture specific? We use the concepts *CommonSensePositiveEntity* and *CommonSenseNegativeEntity* in order to represent this kind of information. For instance, a terrorist would belong to *CommonSenseNegativeEntity* while freedom is an instance of *CommonSensePositiveEntity*. Such knowledge is captured normally by a polarity lexicon. We could merge it into *(Non)SympathyEntity*, but this would confuse personal preferences with broader accepted shared preferences. We keep it separate in order to design a common sense conflict detector and to predict the class of common sense disturbers.

A *common sense conflict* is any situation, where an entity benefits (or suffers) from a situation, but that entity is not worth (does not deserves) it. For instance, if A support terrorism or if A disapproves freedom, a common sense conflict occurs. The actor of it is the *common sense disturber*. We believe that a system that is able to find such sentences could be of great interest for text exploration purposes. Sentences with common sense conflicts might indicate controversial topics, pronounced stance or unusual opinions or at least something that is not desirable from a common sense perspective.

Take these sentences (English translations) found by our system: "Moscow's half-baked attempt to solve the problem only strengthens the radical tendencies" and "These authoritarian forms of dealing with homosexuality are definitely accepted by those conservatives". According to our polarity lexicon, the direct object of "strengthen" ("radical tendencies") receives a positive effect. What makes it a common sense conflict is the bottom-up polarity of "radical tendencies" which is negative. Thus a positive effect (top-down) on a negative entity (bottom-up) establishes a polarity conflict. The same is true for the second sentence where something negative ("authoritarian forms") receives a positive effect (object of "accept") which is a conflict. Articles that contain such conflicts might contain controversial material, highlighting such sentences in a single article might help to focus on the most interesting parts of it.

Currently, we have 8 SWRL rules that establish this inference layer. The goal property is common sense disturber, *cs_disturber*. The first rule is:

```
c1  of_role(?s,?x),aff(?s),factual(?s),
    pfor(?s, ?y),cs_neg(?y)
     -> cs_disturber(?x)
```

If A acts in way that a positive effect (*pfor*) on B takes place, but B is a *CommonSenseNegativeEntity*, *cs_neg* for short, then A is a common sense disturber, *cs_disturber* (e.g. A supports terrorism). However, the degree of negativity of the direct object might play a role. If "The minister supports the rather poor argument", then this might be unwise, but does not touch his common sense integrity. Our polarity lexicon ([Klenner *et al.*, 2014] is designed along the principles of the Appraisal

Theory [Martin and White, 2005]. That is, we distinguish between factually, emotionally or morally positive or negative words. A conflict occurs only if the moral (e.g. crime) or emotional (e.g. fear) dimension is violated, not the factual (i.e. poor decision) one.

Also negated sentences are relevant (rule c2):

```
c2  of_role(?s,?x),neg(?s),factual(?s),
    nfor(?s,?y),cs_neg(?y)
     -> cs_disturber(?x)
```

If A acts in a way that a negative effect on a negative entity does not occur, then A is a common sense disturber.

More complex cases arise if subclause embedding is involved (rule c3):

```
c3  neff(?s,?s2),aff(?s),aff(?s2),
    factual(s2),pof(s2,?y),of_role(?s,?x)
    ->cs_disturber(?x)
```

If a negative effect (*neff*) on a subclause ?s2 is present and the actor of ?s2 receives a positive effect (*pof*), then A, the actor of the matrix verb is a common sense disturber. So if A criticizes that B has helped C to survive, then A is a common sense disturber.

We have carried out a first empirical test on the basis of the German newspaper treebank TüBa-D/Z [Telljohann *et al.*, 2009] comprising 95'500 sentences. Clearly, we cannot expect a huge number of such conflicts, our small lexicon, parsing and verb frame extraction errors are part of the problem. However, the system predicted 64 conflicts from which 31 were - after manual inspection - real conflicts (cf. the two examples from above). So precision is about 50%. Every second sentence proposed by the system does actually point out some interesting charged constellation.

9 Summary

Our model strives to answer the following questions, given a parsed text and the personal preferences of a single user: who benefits (or suffers) from the situations described, what does the text (implicitly) tells about the relationship of the actors involved, which topics does an actor like or dislike and - given all this - what does this implies for the user: who are proponents or opponents of his or hers. Our system is also able to predict situations that – from a common sense perspective – bear controversial or charged content. This could be useful as a new service in the area of media monitoring.

References

[Baroni M., 2009] Ferraresi A. Zanchetta E. Baroni M., Bernardini S. The WaCky Wide Web: A collection of very large linguistically processed Web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226, 2009.

[Davidson, 1967] Donald Davidson. The logical form of action sentences. In Nicholas Rescher, editor, *The Logic of Decision and Action*. 1967.

[Deng and Wiebe, 2015] Lingjia Deng and Janyce Wiebe. Joint prediction for entity/event-level sentiment analysis using probabilistic soft logic models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 179–189, 2015.

[Glimm *et al.*, 2014] Birte Glimm, Ian Horrocks, Boris Motik, Giorgos Stoilos, and Zhe Wang. HermiT: An OWL 2 reasoner. *Journal of Automated Reasoning*, 53(3):245–269, 2014.

[Horridge *et al.*, 2006] Matthew Horridge, Nick Drummond, John Goodwin, Alan Rector, Robert Stevens, and Hai H Wang. The Manchester OWL syntax. In *OWL: Experiences and Directions (OWLED)*, 2006.

[Horrocks and Patel-Schneider, 2004] Ian Horrocks and Peter F. Patel-Schneider. A proposal for an OWL rules language. In *Proc. of the Thirteenth International World Wide Web Conference*, pages 723–731. ACM, 2004.

[Horrocks and Patel-Schneider, 2011] Ian Horrocks and Peter F. Patel-Schneider. KR and reasoning on the Semantic Web: OWL. In John Domingue, Dieter Fensel, and James A. Hendler, editors, *Handbook of Semantic Web Technologies*, chapter 9, pages 365–398. Springer, 2011.

[Karttunen, 2012] Lauri Karttunen. Simple and phrasal implicatives. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics*, SemEval '12, pages 124–131, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[Klenner and Amsler, 2016] Manfred Klenner and Michael Amsler. Sentiframes: A resource for verb-centered german sentiment inference. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may 2016. European Language Resources Association (ELRA).

[Klenner *et al.*, 2014] Manfred Klenner, Michael Amsler, and Nora Hollenstein. Verb polarity frames: a new resource and its application in target-specific polarity classification. In *Proceedings of KONVENS 2014*, pages 106–115, 2014.

[Klenner, 2015] Manfred Klenner. Verb-centered sentiment inference with Description Logics. In *6th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 134–140, September 2015.

[Martin and White, 2005] J. R. Martin and P. R. R. White. *Appraisal in English*. Palgrave, London, 2005.

[Neviarouskaya *et al.*, 2009] Alena Neviarouskaya, Helmut Prendinger, and Mitsuru Ishizuka. Semantically distinct verb classes involved in sentiment analysis. In Hans Weghorn and Pedro T. Isaías, editors, *IADIS AC (1)*, pages 27–35. IADIS Press, 2009.

[Reschke and Anand, 2011] Kevin Reschke and Pranav Anand. Extracting contextual evaluativity. In *Proceedings of the Ninth International Conference on Computational Semantics*, pages 370–374, 2011.

[Telljohann *et al.*, 2009] Heike Telljohann, Erhard W. Hinrichs, Sandra Kübler, Heike Zinsmeister, and Kathrin Beck. Stylebook for the Tübingen treebank of written German (TüBa-D/Z). Technical report, Universität Tübingen, Seminar für Sprachwissenschaft, 2009.

Tell me who you are, I'll tell whether you agree or disagree: Prediction of agreement/disagreement in news blogs

Fabio Celli and **Evgeny A. Stepanov** and **Giuseppe Riccardi**
Signals and Interactive Systems Lab
Department of Information Engineering and Computer Science
University of Trento, via Sommarive 5, Trento, Italy
{fabio.celli,evgeny.stepanov,giuseppe.riccardi}@unitn.it

Abstract

In this paper we address the problem of the automatic classification of agreement and disagreement in news blog conversations. We analyze bloggers, messages and relations between messages. We show that relational features (such as replying to a message or to an article) and information about bloggers (such as personality, stances, mood and discourse structure priors) boost the performance in the classification of agreement/disagreement more than features extracted from messages, such as sentiment, style and general discourse relation senses. We also show that bloggers exhibit reply patterns significantly correlated to the expression of agreement or disagreement. Moreover, we show that there are also discourse structures correlated to agreement (expansion relations), and to disagreement (contingency relations).

1 Introduction

Threaded discussions in on-line social media are asynchronous multiparty conversations that concur to the formation of opinions and shared knowledge which influence decision makers. Bloggers who participate in these conversations usually express their opinions, defend their stances and gain or lose consensus with their text messages. These conversations contain many layers of information such as sentiment [Strapparava and Mihalcea, 2008], humor [Reyes *et al.*, 2012], and Agreement/Disagreement

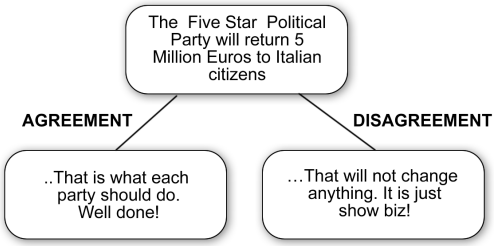


Figure 1: Example of agreement and disagreement.

Relations (henceforth ADRs) [Wang and Cardie, 2014] (see Figure 1 for an example). In this paper we address the problem of extracting ADRs from news blog conversations. There are two possible tasks: ADR detection (finding the messages that contain personal positions) and ADR classification (classifying messages as agreeing or disagreeing with previous messages). Here we address ADR classification and experiment with message, blogger and relational-level features and their combinations.

The paper is structured as follows: in Section 2 we provide an overview of previous work in the field and a definition of ADRs, from Section 3 to 6, we describe the data set, the annotation, the experimental settings and discuss the results.

2 Related Work and Definitions

Previous work on ADRs in asynchronous conversations can be divided into three areas: definition of ADRs, collection and annotation of corpora and prediction of ADRs' polarity.

In [Bender *et al.*, 2011] ADRs are considered

as relationships among bloggers expressed at message level with a post or turn text unit. They collected the AAWD corpus of Wikipedia talk pages and manually annotated with ADRs and authority claims. The reported inter-annotator reliability is k=0.5. In [Walker *et al.*, 2012] ADRs are defined as Quote-Response message pairs and triplets. These pairs and triplets are linked by the structure of the thread, where each message is a reply to its parent and is about the same topic. They collected the IAC corpus [Walker *et al.*, 2012] of political debates in English (about 2700 authors, 11k threads) extracted from *4forums.com* and annotated with ADRs by means of Amazon Mechanical Turk, obtaining inter-annotator reliability of α=0.62. In [Andreas *et al.*, 2012] ADRs are defined between pairs of sentences within messages in a parent/child relation. In their definition, ADRs have a type ("agree", "disagree" or "none") and a mode ("direct" or "indirect", "response" or "paraphrase"). They annotated sentence pairs in a corpus of LiveJournal and Wikipedia with 3 classes ("agree", "disagree", "not applicable"). The reliability between two annotators is k=0.73. In [Celli *et al.*, 2014] ADRs are defined as a function that maps pairs of bloggers' messages to polarity values between 1 ("agree") and -1 ("disagree"). They collected a corpus of news blogs conversations in Italian (CorEA corpus). The reported inter-annotator reliability is k=0.58 on 3 classes ("agree", "disagree", "not applicable") and k=0.87 on 2 classes ("agree", "disagree").

In [Wang and Cardie, 2014], the authors addressed ADRs classification between text segments corresponding to one or several sentences on the IAC and AAWD corpora. The authors observed that it is easier to classify agreement than disagreement in the AAWD corpus, while the contrary is true in the IAC corpus.

3 Dataset

The CorEA corpus [Celli *et al.*, 2014] is used for the experiments throughout the paper. As mentioned in the previous section, the corpus is the collection of news blogs in Italian and consists of asynchronous conversations from 27 news articles on different topics ranging from politics to gossip. The corpus contains 2,887 messages (135K tokens). The average number of messages per conversation is 106.4.

The corpus has been labeled by two annotators with three labels: "agreement", "disagreement" and "not applicable" (henceforth "NA"). Messages are annotated with a "NA" label, if they satisfy one or both of the following conditions: a) **message is not clear**, if the annotator cannot find or commit to the relation between parent and child messages (e.g. the child message contains only URLs or is not referred to its parent); b) **message contains mixed agreement**, if in the child message there are conflicting or ambiguous cues triggering agreement and disagreement. This includes cases such as conflicting opinions in the child message about one or more statements in the parent message. If the message does not fall under the cases specified above, the ADR in the child message is evaluated with respect to the parent as "agree" (1) or "disagree" (-1). The distribution of labels in the corpus is 31% agreement, 34% disagreement, and 35% NA.

4 Features

For the experiments on ADR classification, we exploit the features already present in the data and enriched them with new features at the level of messages, bloggers and parent-child relations (relational features).

Message-level Features (107). *Discourse Features* (8) are frequency counts and ratios (% from total) of the four top-level relation senses from Penn Discourse Treebank (PDTB) [Prasad *et al.*, 2008]: Comparison, Contingency, Expansion, and Temporal. They are extracted for explicit discourse relations (signaled by connectives such as *but*, *however*, *when*, etc.) using lexical context classifier of [Riccardi *et al.*, 2016]; and a connective sense classifier trained on Italian LUNA Corpus [Dinarelli *et al.*, 2009].*Sentiment Polarity Features* (2) are text-length normalized sums of the polarized words extracted using OpeNER lexicon (http://www.opener-project.eu/), and their discretisation into positive, neutral, and negative classes. *Stylometric Features* (97) are basic text statistics (4) such as word count, vocabulary size, average word length; frequency-based features (2) such as frequency of hapax legomena; measures of lexical richness (16) based on word count, vocabulary size, and word-frequency spectrum such as mean word frequency, type-token ratio, entropy, Guiraud's R, Honore's H, etc. [Tweedie and Baayen, 1998]; and word length ratios (30) for 1-30 character long words. The feature set also includes character-based ratios (45) for character classes (e.g. punctuation,

white space, etc.) and individual characters (e.g. '!', 'a', etc.). Additionally, we include the number of message likes and replies (2).

Relational-level Features (4). These features are generated using child and parent messages (or the article as parent). They include word2vec [Mikolov *et al.*, 2013] cosine similarity between parent and child messages, boolean feature to indicate whether a parent is an article or another message, and two boolean features for matches and mismatches between topics and sentiment polarities expressed in two messages.

Blogger-level Features (22). Blogger-level features are the personality types (5), self-assessed blogger mood priors (5); the aggregation (sums and averages) of the message-level discourse (8) features; blogger's stance (1), and blogger's topic per message ratio (1). Personality types are defined by the Five Factor Model: extroversion, emotional stability/neuroticism, agreeableness, conscientiousness, openness to experience. These features have been automatically predicted exploiting linguistic cues from the collection of all messages of single bloggers. The accuracy of the prediction, evaluated on an Italian Facebook dataset [Celli, 2013], is 65%. Mood priors encoded in CorEA are: indignation, disappointment, worry, amusement and satisfaction. Stance is the sum of the polarity of messages of a blogger.

5 Experiments and Results

As it was already stated, in this paper we address the problem of classification of ADRs as pairs of parent-child messages being in agreement or disagreement relation. Thus, the problem is case as a binary classification task; as opposed to the 3-way classification including "NA" relations or a two-step hierarchical ADR detection-classification task. For the experiments, we have balanced the data and partitioned it into training and testing as 66% and 33%. Since some blogger level features are aggregations of message-level features, the data was split by alphabetically sorting the messages by bloggers' names. Support Vector Machine classifier with linear kernel from Weka is used as learning algorithm.

The results on ADR classification using message, blogger-level and relational-level features and their combinations are reported in Table 1. Similar to the observation of [Wang and Cardie, 2014] for English on AAWD, we observe that classification performance for agreement is higher than disagreement for Italian as well. With respect to feature groups, we observe that blogger and relational features outperform the bag of words baseline. The best performance is obtained using relational feature only, followed by the blogger-level features. In order to evaluate the contributions of individual features, we have performed correlation analysis. Table 2 reports the ranking of the features highly correlated with agreement and disagreement labels. We also observe that bloggers who reply to the article tend to agree with its content, and this can be seen as a result of the quality of the information of the article and the credibility of news. In debate corpora (e.g. IAC) such a tendency is not observed. Moreover, bloggers that get more replies are the ones that disagree the most with others, and this can be explained with the fact that disagreement generates a debate. It is also interesting to note that extroversion is correlated to disagreement and conscientiousness to agreement. With respect to discourse structure, we observe that contingency and comparison rela-

settings	**agree (F1)**	**disagree (F1)**	**both (acc)**
majority baseline	0.500	0.500	0.500
bag of word baseline	0.550	0.624	0.590
message	0.555	0.554	0.550
blogger	0.634	0.568	0.601
relational	**0.726**	**0.684**	**0.705**
message+blogger	0.618	0.560	0.589
message+relational	0.711	0.675	0.693
blogger+relational	**0.726**	**0.684**	**0.705**
all	0.659	0.629	0.644

Table 1: Result of the classification of ADRs using different combinations of message features, blogger features and relational features. We use 66% training, 33% test split a Support Vector Machine as classifier (Weka SMOreg), F1 and accuracy (acc) as evaluation metrics.

Corr	**feat. type**	**feature**	**Class**
0.418	relational	article as parent	A
0.265	blogger	reply ratio	D
0.205	blogger	topic-message ratio	A
0.183	blogger	expansion	A
0.147	message	ratio of 2-char words	D
0.146	blogger	conscientiousness	A
0.127	message	! marks ratio	D
0.126	blogger	contingency	D
0.121	blogger	extroversion	D
0.106	blogger	comparison	D
0.105	message	replies count	D

Table 2: Ranking of the features highly correlated with agreement (A) and disagreement (D) (Pearson's correlation with $p - value < 0.001$).

tions tend to be used for expressing disagreement, while expansion relations are mainly used to express agreement. Moreover, this is complemented by the fact that bloggers in agreement with others tend to address more topics in a single message. Among the other observations, we notice that exclamation marks and short words are strong cues for disagreement.

6 Conclusion

In this paper we addressed the problem of classification of message-pairs from online conversations into agreement and disagreement relations. We have demonstrated that blogger-level and relational-level features outperform the message-level features, such as sentiment polarity and style. Through correlation analysis we have studied how agreement and disagreement relations are expressed. We have observed that there are discourse structures underlying the expression of agreement and disagreement relations in social media. The methodology presented in this paper is useful for the automatic analysis of online social media conversations. The future work includes the detection of agreement/disagreement relations and their exploitation for conversation summarisation.

Acknowledgements

The research leading to these results has received funding from the European Union – Seventh Framework Programme (FP7/2007–2013) under grant agreement 610916: SENSEI.

References

[Andreas *et al.*, 2012] Jacob Andreas, Sara Rosenthal, and Kathleen McKeown. Annotating agreement and disagreement in threaded discussion. In *LREC*, 2012.

[Bender *et al.*, 2011] Emily M Bender, Jonathan T Morgan, Meghan Oxley, Mark Zachry, Brian Hutchinson, Alex Marin, Bin Zhang, and Mari Ostendorf. Annotating social acts: Authority claims and alignment moves in wikipedia talk pages. In *Proceedings of WLSM*, pages 48–57. ACL, 2011.

[Celli *et al.*, 2014] Fabio Celli, Giuseppe Riccardi, and Arindam Ghosh. Corea: Italian news corpus with emotions and agreement. In *Proceedings of CLIC-it 2014*, pages 98–102, 2014.

[Celli, 2013] F Celli. *Adaptive Personality recognition from Text*. Lambert Academic Publishing, 2013.

[Dinarelli *et al.*, 2009] Marco Dinarelli, Silvia Quarteroni, Sara Tonelli, Alessandro Moschitti, and Giuseppe Riccardi. Annotating spoken dialogs: from speech segments to dialog acts and frame semantics. In *Proceedings of EACL Workshop on the Semantic Representation of Spoken Language*, Athens, Greece, 2009.

[Mikolov *et al.*, 2013] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *HLT-NAACL*, 2013.

[Prasad *et al.*, 2008] R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi, and B. Webber. The penn discourse treebank 2.0. In *Proc. of LREC*, 2008.

[Reyes *et al.*, 2012] Antonio Reyes, Paolo Rosso, and Davide Buscaldi. From humor recognition to irony detection: The figurative language of social media. *Data & Knowledge Engineering*, 74:1–12, 2012.

[Riccardi *et al.*, 2016] Giuseppe Riccardi, Evgeny A. Stepanov, and Shammur Absar Chowdhury. Discourse connective detection in spoken conversations. In *Proc. of ICASSP*, 2016.

[Strapparava and Mihalcea, 2008] Carlo Strapparava and Rada Mihalcea. Learning to identify emotions in text. In *Proceedings of the 2008 ACM Symposium on Applied Computing*, SAC '08, pages 1556–1560, New York, NY, USA, 2008. ACM.

[Tweedie and Baayen, 1998] Fiona J. Tweedie and R. Harald Baayen. How variable may a constant be? measures of lexical richness in perspective. *Computers and the Humanities*, 32(5):323–352, 1998.

[Walker *et al.*, 2012] Marilyn A Walker, Jean E Fox Tree, Pranav Anand, Rob Abbott, and Joseph King. A corpus for research on deliberation and debate. In *LREC*, 2012.

[Wang and Cardie, 2014] Lu Wang and Claire Cardie. Improving agreement and disagreement identification in online discussions with a socially-tuned sentiment lexicon. *ACL 2014*, page 97, 2014.

Creation, Visualization and Edition of Timelines for Journalistic Use

Xavier Tannier
LIMSI, CNRS, Univ. Paris-Sud,
Université Paris-Saclay
F-91405 Orsay, FRANCE
xavier.tannier@limsi.fr

Frédéric Vernier
LIMSI, CNRS, Univ. Paris-Sud,
Université Paris-Saclay
F-91405 Orsay, FRANCE
frederic.vernier@limsi.fr

Abstract

We describe in this article a system for building and visualizing thematic timelines automatically. The input of the system is a set of keywords, together with temporal user-specified boundaries. The output is a timeline graph showing at the same time the chronology and the importance of the events concerning the query. This requires natural language processing and information retrieval techniques, allied to a very specific temporal smoothing and visualization approach. The result can be edited so that the journalist always has the final say on what is finally displayed to the reader.

1 Introduction

Timelines are a natural way to describe series of related events in a compact manner, and journalists use them a lot. However, writing and maintaining such timelines, as well as building a comprehensive visualization, requires a considerable amount of human effort.

For this reason, automatic timeline summarization (TS) has known a wide interest in the last past years. TS is generally seen as a special case of multi-document summarization. For that matter, multi-document summarization systems have been used to generate timelines, and focus on the selection of the most representative sentences in an already time-stamped corpus [Yan *et al.*, 2011; Chieu and Lee, 2004; Tran *et al.*, 2015b]. Some previous work have focused on extracting salient dates before selecting the description of events corresponding to these dates [Tran *et al.*, 2013; 2015a; Kessler *et al.*, 2012; Nguyen *et al.*, 2014].

The final output of these systems is generally made of the k top ranked events, presented in chronological order. The visualization can then be obtained with traditional librairies such as TimelineJS, SIMILE, TimeGlider, vis.js (see Figure 1). The information concerning the rank and the weight of the events is only used for selecting the top k, and is then lost.

We argue that readable timelines (or chronologies) should present first an overview with the most important events, but also let the reader discover intermediate events at will. A timeline must certainly be followed along a temporal axis, but a feedback of the importance of the events should also be displayed.

In this paper, we follow [Nguyen *et al.*, 2014] and describe a system taking a set of keywords as an input, producing an output that is not a constrained summary or list of events, but a weighted list of dates, together with a description of the event that occurred at each date. We first focus on how the articles are processed in order to rank the dates, and especially on how the events are time-stamped. We show that considering only the article publication date does lead to shifted peaks, and then to irrelevant timelines. For this reason, we use a temporal normalization of texts to adjust the peaks. Then, we choose the best article headline related to the date and topic, as an event description.

Finally, we present a visualization tool specially dedicated to this system, where all the extracted events in the considered time span can be shown, and where the importance of the events is symbolized by a time-series graph filtered through event-specific smoothing functionalities.

The system is demonstrated on French data.

2 System

The first step of our approach can be seen as a task of "date extraction". Our system extracts a maximum of temporal information and uses only this information to detect salient dates for the construction of event timelines. Then, textual content is used for selecting the description of each event. Finally, an original data visualization is proposed.

2.1 Event Extraction

Figure 2 shows the general architecture of the system:

① The system Heideltime [Strötgen and Gertz, 2013; Moriceau and Tannier, 2014] is used to normalize temporal expressions in the texts. Absolute (e.g. "*January 6, 2016*") and relative dates (e.g. "*on Friday*") are turned into a `YYYY-MM-DD` common format (see examples in Figure 3). This allows us to link event to specific dates, instead of relying on the document creation time.

② The corpus is indexed by the Solr search engine[1], where one document per sentence is created, considering only sentences containing a normalized date. Each sentence

[1] `http://lucene.apache.org/solr/`

Figure 1: An existing, manually produced timeline on the French presidential race (TimelineJS).

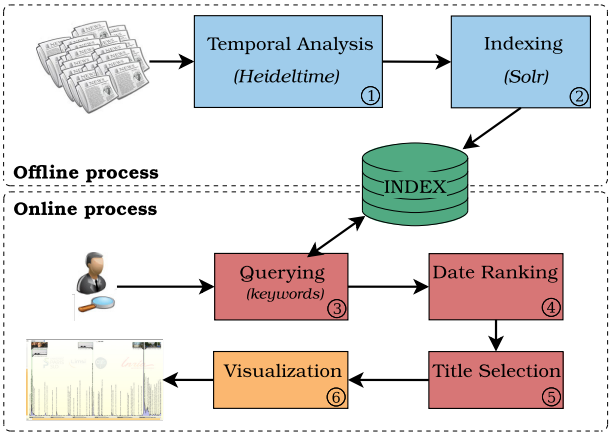


Figure 2: System overview.

At least 129 people died after a series of violent incidents around Paris, France, on ***Friday 13 November 2015****.*
The attacks in Paris on the night of ***Friday 13 November*** *left 130 people dead and hundreds wounded.*
At least 128 people were killed in shootings and explosions in Paris ***late Friday***
Attacks such as the one in Paris ***three days ago*** *cannot obliterate our desire to understand*

Figure 3: Examples of sentences refering to November 13 events with absolute or relative dates. The normalized is "2015-11-13" for all the sentences.

is indexed together with the dates and the title of its article, using stemming.

③ At query time, documents are retrieved from the index, without any number limit.

④ Dates are extracted from the documents and weighted according to the number of occurrences of the date in the retrieved documents. Thus we obtain a plot where each peak corresponds to an "important" date. This is why considering dates inside the text instead of the document creation time is important: using document creation time gives us a measure of the mediatic response to events, making the peaks match with the days after the events. Retiming the events w.r.t. the dates specified in the text allows to reposition the peaks in front of the actual date of the events (see an illustration at Figure 4).

⑤ We then need to associate a textual description to each event. This is done by collecting the more important words for each date with a classical tf.idf:

$$tf.idf(w,d) = tf(w,d) \log \frac{N}{df(w)}$$

where $tf(w,d)$ is the frequency of word w in all sentences containing the date d, $df(w)$ is the frequency of word w in the entire corpus, and N is the total number of documents in the corpus. For each date d, the 20 words having the highest weight are used to query the Solr index again and to select the top article published at this date d. The description of the day event is then the headline and a picture (if any) of this article.

⑥ Visualization is described at next Section.

2.2 Visualization Tool

Figure 5 shows an example of graph produced by the system. On top, the most relevant events are presented (headline and picture from the selected article for the specific day). At bottom, the graph is displayed along the same temporal axis. The graph represents the weights of each day as calculated at step ④. However, like all measures representing a human activity, these weights lead to a very noisy graph. We provide then a smoothing function to make the graph more readable to the user. A traditional Gaussian blur can be added and controled to obtain a smoother curve, but it also shifts the maxima to the right. Therefore, it would lose the temporal signature of the burst and decoy model (Descending Triangle Reversal [Hochheiser and Shneiderman, 2001]). In consequence, we added another smoothing functionality based on Bilateral Filtering [Paris and Durand, 2008] to preserve the discontinuity at event burst. However it does not match our burst and decoy model since days just before a burst are raised by the following days if decoy happens into the kernel size. We refined then the Bilateral Filtering by accounting only past events in the smoothing function. This function is then:

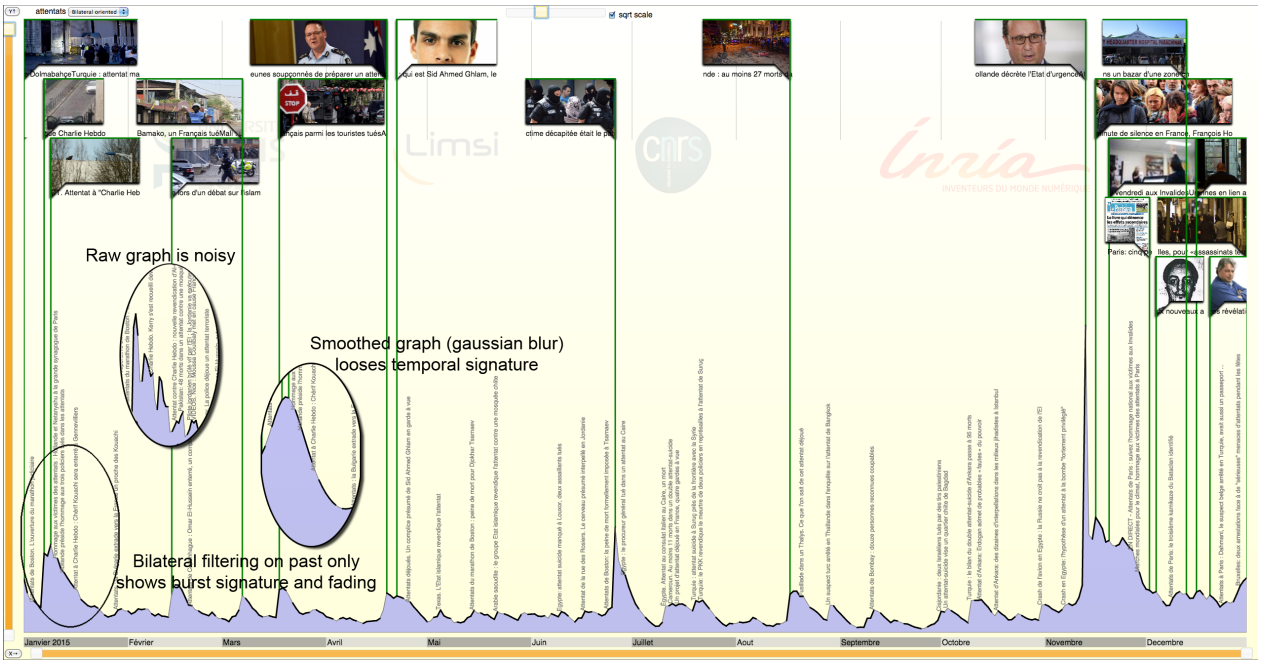


Figure 5: Visualization for the query "attentats" ("*attacks*") in 2015.

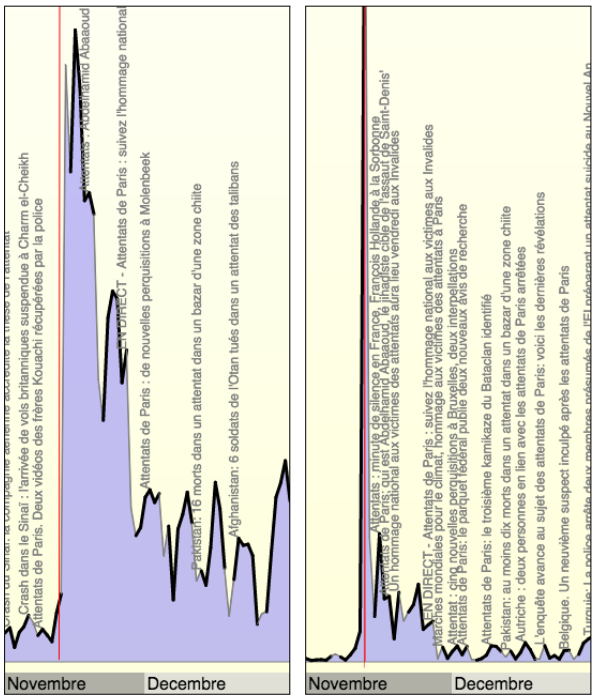


Figure 4: Raw graphs on query "attacks" in November and Decembre 2015, with time weights given by the document creation time (left) or by the temporal normalization (right). November 13 is represented by the red vertical line.

$$w'(d) = \frac{\sum_{i=-2\rho}^{0} w(d+i) \times e^{\frac{-i^2}{\rho^2}} \times e^{-\frac{(w(d)-w(d+i))^2}{\sigma^2}}}{\sum_{i=-2\rho}^{0} e^{\frac{-i^2}{\rho^2}} \times e^{-\frac{(w(d)-w(d+i))^2}{\sigma^2}}}$$

where $w(d)$ is the initial weight as described in previous section, d is the considered day, ρ an integer parameter and σ a real parameter. ρ and σ represent the extension of the neighborhood (ρ^2 is the variance of the Gaussian function) and can be modified by the user through sliders, for more or less smoothing. Default values are $\rho = 2$ and $\sigma = 0.1$. It produces the nicely readable graph of Figure 5 instead of the ones circled in the same Figure.

Even if a smoothing is necessary, we still aim at obtaining strong and sharp peaks when important events occur. Instead of using a pure burst model as in Kleinberg [2002] or Zhu and Shasha [2003], which have already been applied to media content [Xie *et al.*, 2013; Takahashi *et al.*, 2012], we prefer using our refined Bilateral Filtering with a decreasing threshold detection. We use the double gaussian Kernels of the Bilateral Filtering as the aggregate function F of the Shasha Model [Zhu and Shasha, 2003]. The article at highest burst is selected then removed from the time serie. This process is repeated until the timeline is filled with the targeted number of selected events.

Selected articles are displayed with both an image and the title of the article underneath. The system crops the image to a flag shape to highlight the temporal nature of events. The flag shape is attached to the graph with a line from the triangle. When two events happen very close to each other the flags can float on different sides of their pole (first event is dis-

played on the left to avoid overlapping). When more than two events occur very close we chose to display them arbitrarily on different tracks. As we process selected events from highest to lowest ranked, most important events are displayed on the top track and least important ones appear underneath. The smaller, not selected peaks display vertically the titles of the selected articles for these days.

The users of the system (i.e. journalists) can zoom on both axes by using the range sliders at bottom and at left of the combined graph. A temporal legend always display time ticks at bottom of the graph. The users can interactively rearrange events since the layout mechanism can fail to optimize screen real estate. Users can flip the flags on both side, change track (up or down) of an event and switch between two sizes (big or small). Furthermore, users can downgrade an event (and make it a smaller peak) or upgrade a smaller peak by double clicking on an vertical title above the graph. This makes the result fully editable, so that the journalist has the final say on what is displayed.

3 Discussion

3.1 Limits

The first important limit of the system is that its event granularity is fixed. This leads to two main issues: 1/ The tool is not able to detect more than one event per day. 2/ A macro-event that would last more than one day (e.g. a conference) could not be extracted nor vizualized.

Workarounds have been considered [Nguyen *et al.*, 2014] but tend to reduce the overall accuracy of the system. Our further work will focus on this issue.

The definition of an event "importance" is also open to question. In this paper we considered to the importance depends only on repetition. Other factors have been studied and applied with learning-to-rank approaches [Kessler *et al.*, 2012], and should be integrated into this system.

Finally, the process requires a large number of search engine queries, which makes it time-consuming. A first query returns a potentially high number of documents; then, one query per day in the time span is run to select the best article. Even if they can easily be parallelized, all these queries make the entire process quite heavy[2].

3.2 Adaptation to Other Languages

This study has been achieved on a French dataset. Only two steps are language-dependent and need little adaption to another language:

- Tokenization and stemming (widely available in many languages)
- Temporal normalization. Heideltime is available in 13 languages at the time of writing, and other tools may be existing for other languages.

The next step that is now being conducted consists in an evaluation with our journalist partners, and considers both the accuracy of the timeline and the ergonomics.

[2]Within a single thread on a simple server, about one minute for a one-year query on a popular subject as "attacks".

References

[Chieu and Lee, 2004] Hai Leong Chieu and Yoong Keok Lee. Query based event extraction along a timeline. In *Proceedings of the 27th ACM SIGIR conference*, 2004.

[Hochheiser and Shneiderman, 2001] H. Hochheiser and B. Shneiderman. Visual Specification of Queries for Finding Patterns in Time-Series Data. Technical Report CS-TR-4326, University of Maryland, 2001.

[Kessler *et al.*, 2012] Rémy Kessler, Xavier Tannier, Caroline Hagège, Véronique Moriceau, and André Bittar. Finding Salient Dates for Building Thematic Timelines. In *Proceedings of the 50th Annual Meeting of the ACL*, 2012.

[Kleinberg, 2002] J. Kleinberg. Bursty and Hierarchical Structure in Streams. In *Proceedings of the 8th ACM SIGKDD Conference*, 2002.

[Moriceau and Tannier, 2014] Véronique Moriceau and Xavier Tannier. French Resources for Extraction and Normalization of Temporal Expressions with HeidelTime. In *Proceedings of the 9th LREC Conference*, 2014.

[Nguyen *et al.*, 2014] Kiem-Hieu Nguyen, Xavier Tannier, and Véronique Moriceau. Ranking Multidocument Event Descriptions for Building Thematic Timelines. In *Proceedings of the 30th Coling Conference*, 2014.

[Paris and Durand, 2008] Kornprobst P. Tumblin J. Paris, S. and F. Durand. A Gentle Introduction to Bilateral Filtering and its Applications. In *Proceedings of the 42nd International SIGGRAPH Conference*, 2008.

[Strötgen and Gertz, 2013] Jannik Strötgen and Michael Gertz. Multilingual and Cross-domain Temporal Tagging. *Language Resources and Evaluation*, 47(2), 2013.

[Takahashi *et al.*, 2012] Y. Takahashi, T. Utsuro, M. Yoshioka, N. Kando, T. Fukuhara, H. Nakagawa, and Kiyota Y. Applying a Burst Model to Detect Bursty Topics in a Topic Model. In *Proceedings of JapTAL*, 2012.

[Tran *et al.*, 2013] G. Tran, M. Alrifai, and D. Q. Nguyen. Predicting Relevant News Events for Timeline Summaries. In *Proceedings of WWW Conference*, 2013.

[Tran *et al.*, 2015a] G. Tran, E. Herder, and K. Markert. Joint Graphical Models for Date Selection in Timeline Summarization. In *Proceedings of the 53rd ACL*, 2015.

[Tran *et al.*, 2015b] Giang Tran, Mohammad Alrifai, and Eelco Herder. Timeline Summarization from Relevant Headlines. In *Proceedings of the 37th ECIR*, 2015.

[Xie *et al.*, 2013] F. Xie, W.and Zhu, J. Jiang, and Lim E.P.and Wang K. TopicSketch: Real-time Bursty Topic Detection from Twitter. In *Proceedings of IEEE 13th ICDM*, 2013.

[Yan *et al.*, 2011] Rui Yan, Liang Kong, Congrui Huang, Xiaojun Wan, Xiaoming Li, and Yan Zhang. Timeline Generation through Evolutionary Trans-Temporal Summarization. In *Proceedings of the 2011 EMNLP*, 2011.

[Zhu and Shasha, 2003] Y. Zhu and D. Shasha. Efficient elastic burst detection in data streams. In *Proceedings of the Ninth ACM SIGKDD* , 2003.

Towards Semantic Story Telling with Digital Curation Technologies

Julian Moreno Schneider, Peter Bourgonje, Jan Nehring, Georg Rehm, Felix Sasaki, Ankit Srivastava
DFKI GmbH, Language Technology Lab, Alt-Moabit 91c, 10559 Berlin, Germany
dkt@dfki.de

Abstract

We develop a system that aims at generating stories or, rather, potential story paths, based on the semantic analysis of multiple source documents (including news articles) using template-filling. The final system will be realised by additional methods, also taking specific domains and topics into account. For the processing we use NLP methods such as named entity recognition, we also use a triple store and classic document indexing modules. The analysis information is filtered, rearranged and recombined to fit the respective template. The system's use case is to support knowledge workers (journalists, editors, curators etc.) in their tasks of processing large amounts of (incoming) documents, to identify important entities, relationships between entities and to suggest individual story paths between entities, eventually to come up with more efficient processes for content curation.

1 Introduction and Context

Journalists have to cope with huge amounts of incoming information that need to be scanned, skimmed, contextualised, evaluated and, eventually, further processed into a new piece of news, blog post, or longer article. The demand for tool support is extremely high. Many journalists and online creators are under a lot of pressure as they are expected to produce as many pieces as possible in less and less time.

However, it is not only journalists who have a growing demand for semantic tools to help them with data processing (in terms of efficiency, breadth, depth, scope etc.), ascertaining what is important and relevant, maybe even genuinely new, surprising and eye-opening. In addition to journalists who work for traditional or online news outlets (incl. blogs, newspapers, radio and tv stations etc.), there are many other job profiles that have to cope with a rather high volume of incoming news or, on a more general level, content that need to be processed in a rather short amount of time in order to produce something new – let us call this group "knowledge workers". These can be, among others, authors and creatives who work in an agency specialised on building information portals: the client provides a smaller or larger amount of data, information, documents, and pictures that now needs to be processed into an interactive website. Second example: creatives who work in an agency that specialises on conceptualising and producing museum exhibitions and showrooms. On a regular basis, these teams face the challenge of becoming experts on a completely new topic basically overnight, when they are confronted with a huge pile of highly domain-specific information that needs to be transformed into an exhibition (or into a convincing pitch to actually get the contract for the production of the new exhibition).

The common ground of the different tasks and challenges described above is the *curation of digital content*. In our project Digital Curation Technologies[1] we collaborate with four SME companies that cover the different use cases and sectors mentioned above, including journalism [Rehm and Sasaki, 2016]. The goal of our project is to design and to build language and knowledge technologies that support the knowledge workers and that help them to become more efficient by delegating routine tasks to the machine with a focus on use-case specific text documents (we currently work on data sets provided by our four SME partners) so that the knowledge workers can concentrate on their core tasks, i. e., producing a story or document that is based on a specific genre or text type (a news piece, an exhibit, a tv news report etc.) and that relies on facts and figures contained in a heterogeneous collection of content.

Among the tools that we develop and integrate into our emerging Platform for Digital Curation Technologies are semantic story telling, named-entity recognition, entity linking, temporal analysis, machine translation, summarisation, classification and clustering [Bourgonje *et al.*, 2016]. We currently focus upon providing RESTful APIs to our SME partners that provide basic functionalities that can already now be integrated into their own in-house systems. In addition, we work on the more complex, longer-term idea of designing and implementing a system for Semantic Story Telling. This system will eventually be able to take a large amount of documents, extract entities and relations between entities, also extract temporal information and relationships, automatically produce a hypertext view of the document cluster in order to enable knowledge workers quickly and efficiently to familiarise themselves with the document collection (i. e., with a new domain or a new topic). We also experiment with the

[1] DKT, see http://www.digitale-kuratierung.de for more details.

idea of automatically generating story paths through this hypertext cluster that can then be used as the foundation of a new piece of content. In a later stage of the project we plan to augment our technologies with state of the art big data systems in order to be able to process high volumes of news data in motion.

The paper is structured as follows: Section 2 discusses related work. Section 3 presents the current architecture of our system. An initial evaluation is described in Section 4. Section 5 concludes the article.

2 Related Work

Our Semantic Story Telling approach is rooted in and influenced by several different approaches in the area of text understanding and generation. [Rumelhart, 1975] was among the first to break down texts of specific types into smaller components by introducing the notion of story grammars that provide established conventions with regard to structure, contents and expectations. Multiple authors developed these concepts further, see, for example, [Orlikowski and Yates, 1994], who define a genre as "a distinctive type of communicative action, characterised by a socially recognised communicative purpose and common aspects of form. The communicative purpose of a genre is not rooted in a single individuals motive for communicating, but in a purpose that is constructed, recognised, and reinforced within a community". [Mann and Thompson, 1988] introduced Rhetorical Structure Theory as a means of describing and specifying the rhetorical relationships between parts of a text. [Rehm, 2002] combined several of these approaches into a system for the automatic identification of different web genres. [Rehm, 2005] provides a comprehensive overview of the literature.

In our current, early prototype implementation we combine several different NLP and IR methods with the goal of providing curation technologies to knowledge workers in different sectors. While many approaches, for example, in the context of museums or libraries, focus upon digital museums (i. e., form and presentation) [Y.-C. Li, 2012] or digital libraries [Meghini and Bartalesi, 2014], we focus upon supporting the actual internal procedures and processes that are used for preparing a real or digital museum (through semantic analysis of the content to be curated, automatically creating metadata etc.). We concentrate on the discoverability of curated content and establishing semantic relations between concepts (i. e., entities or relations) to improve understanding of the subject of research. We also include external ontologies and linked data sources. The system described by [Lewis *et al.*, 2014] is similar to our approach but it is targeted at the localisation industry and uses different data models.

3 System Overview

The current Digital Semantic Storytelling System (DS3) prototype involves several components and two main processing phases (see Figure 1). The user manually selects a story template (Section 3.1) that is then processed and filled (Section 3.2). This process is based on a semantic layer [Bourgonje *et al.*, 2016] that is constructed on top of the respective document collection (Section 3.3).

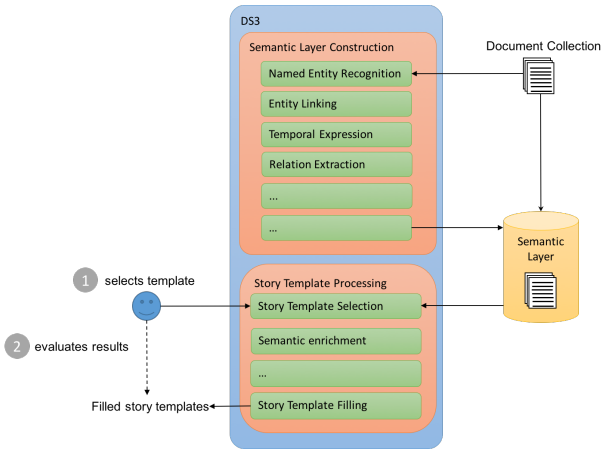


Figure 1: The architecture of the DS3 prototype

3.1 Template Definition and Selection

The process starts with the user selecting the template that applies to and best fits the particular topic and domain of the document collection. The prototype currently contains two templates. Templates are defined in a Template Pattern File (TPT). A TPT file contains the title of the template followed by information about its structure and content (the individual fields of the template). A field definition consists of several columns, the number of columns depends on the type of the field (defined in the second column). A missing value in a column is represented by a single question mark. We currently define the following columns:

1. **Name:** the name of this field
2. **Type:** the type of the field; currently there are two possible field types: single (an entity with an associated URL) and triple (used for storing relations)
3. **Required:** boolean (field required: yes/no)
4. **Subject:** required if field type is triple
5. **Predicate:** required if field type is triple
6. **Object:** required if field type is triple

For the full system we need to significantly extend not only the conceptual specification of templates but also the currently implemented set of templates. For now we have included two templates (*Biography* and *News*). In the following, we focus upon the News template.

- **Biography:** *maincharacter*, *dateofbirth*, *placeofbirth*, *dateofdeath*, *placeofdeath*, *placeofresidence*, *relationship* (*maincharacter* field is single type, the other fields are of type triple).
- **News:** *mainfact*, *locations*, *persons*, *organisations*, *times/dates* (*mainfact* field is single type, the other fields are of type triple).

3.2 Template Filling

The template defines – in the current prototype still in a rather loose sense – the structure of a story. In the Template Filling

phase, analysis information provided by the Semantic Layer (see Figure 1) is used to fill the template. In DS3 we use the Sesame framework for semantic storage. The Semantic Layer contains information that originates in external ontologies (DBPedia, German National Library, verb ontologies etc.) as well as several NLP methods.

3.3 Semantic Layer Construction

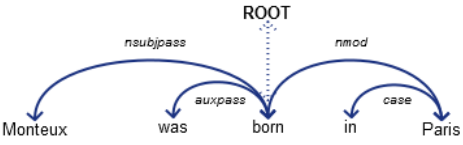
The semantic analysis consists of a pipeline that combines named entity recognition (NER), entity linking, temporal analysis and simple relation extraction. The modules in the pipeline are connected through the NLP Interchange Format (NIF) [Sasaki *et al.*, 2015]. Each analysis takes either plain text or NIF as input and outputs NIF, in which the additional semantic information is stored as annotations.

Our NER module is based on OpenNLP. The approach combines models (if training data is available) and dictionaries (if domain-specific data such as compiled word lists or gazetteers, provided by our SME partners, is available). For every recognised entity we attempt to retrieve a URI in either a domain-specific ontology or DBPedia. If we successfully retrieve a URI, we proceed to use type-specific SPARQL queries to retrieve additional information (e. g., latitude and longitude points for locations, date of birth and death for persons).

The Temporal Analysis module is based on a regular expression grammar. Recognised expressions are resolved to a fully-specified format. For underspecified dates, an anchor date is used for normalisation. This is either the creation date of the document (if available) or another, previously normalised temporal expression. The final annotation is always a range, allowing the inclusion of more specific dates in less specific dates (i. e., we can recognise that, e. g., "13-04-2015" is part of "April of 2015"). The module is rule-based and currently works for English and German.

Explicitly encoded or annotated relations are an important prerequisite for attempting to generate story paths over a set of concepts or entities. In terms of relation extraction we currently experiment with the Stanford CoreNLP dependency parser [Manning *et al.*, 2014]. Our current approach is to extract subject-relation-object triples for which the governing node is a verb. The dependency that has a subject type relation is taken as the subject and the dependency having an object type relation is taken as the object. We subsequently filter for triples for which the subject and the object are named entities for which a URI has been retrieved. These are stored in an internal ontology. This results in a collection of relation triples that we can use to fill templates. Figure 2 shows an example dependency graph for the sentence "Monteux was born in Paris" and the corresponding NIF annotation; by using token indices we can combine the two to arrive at the triple *[http://d-nb.info/gnd/122700198, born, http://www.geonames.org/2988507]*, which can fill the birth place slot in a biography template. A drawback of this approach we found is that the number of relations that were extracted are relatively limited. This is due to the requirement that both the subject and the object of the triple must be governed by the same verb node, and the filtering step, where we keep only those relation triples for which both the subject and the object was and could be recognized as an entity and also resolved to a URI. We want to ensure connectability with external ontologies, thus keep the filtering in place (e. g., a relation triple like *[Mary, met, John]* where we have no further information regarding *Mary* or *John* in the form of a URI in an ontology is currently only of limited use for our application). To increase the number of useful relations in future versions of our relation extraction component, we are experimenting with finding the lowest governing node that is a verb that connects the subject and object nodes in the graph. This verb will then be taken as the type of relation. In addition, we will look into other, dedicated and more sophisticated tools and approaches for relation extraction. Furthermore, not only do we have to identify relations between entities or concepts and their specific features or individual characteristics, but also relations with regard to, among others, coreference of entities, relations between different parts of a document, relations between the same instances of concepts mentioned in multiple documents, to name just a few.

The semantic annotations mentioned above constitute the semantic layer on top of the document collection that is being processed when filling a template.




```
<http://dkt.dfki.de/documents/#char=0,25>
 a                    nif:RFC5147String, nif:String, nif:Context ;
 nif:beginIndex       "0"^^xsd:nonNegativeInteger ;
 nif:endIndex         "25"^^xsd:nonNegativeInteger ;
 nif:isString         "Monteux was born in Paris"^^xsd:string .

<http://dkt.dfki.de/documents/#char=20,25>
 a                    nif:RFC5147String, nif:String ;
 nif:anchorOf         "Paris"^^xsd:string ;
 nif:beginIndex       "20"^^xsd:nonNegativeInteger ;
 nif:endIndex         "25"^^xsd:nonNegativeInteger ;
 nif:entity           <http://dkt.dfki.de/ontologies/nif#LOC> ;
 nif:referenceContext <http://dkt.dfki.de/documents/#char=0,25> ;
 itsrdf:taIdentRef    <http://www.geonames.org/2988507> .

<http://dkt.dfki.de/documents/#char=0,7>
 a                    nif:RFC5147String, nif:String ;
 nif:anchorOf         "Monteux"^^xsd:string ;
 nif:beginIndex       "0"^^xsd:nonNegativeInteger ;
 nif:endIndex         "7"^^xsd:nonNegativeInteger ;
 nif:entity           <http://dkt.dfki.de/ontologies/nif#PER> ;
 nif:referenceContext <http://dkt.dfki.de/documents/#char=0,25> ;
 itsrdf:taIdentRef    <http://d-nb.info/gnd/122700198> .
```

Figure 2: Dependency graph for "Monteux was born in Paris" and the corresponding NIF document

An initial proof-of-concept of the DS3 semantic information retrieval module is available online.[2] Figure 3 shows the filled in *News* story template that was generated for the user-selected initial concept "Erich Mendelsohn". The system provides concepts related to the initial concept. For each related entity, subtrees are built. Our goal with this approach and system is, ultimately, to provide a tool that knowledge workers can use not only to explore a semantic space in an interactive way but to get support for the identification of interesting

[2] http://dev.digitale-kuratierung.de/ds3/

story paths in a potentially huge concept space that the knowledge worker is not familiar with. We will also include a feedback mechanism so that the user can up-vote or down-vote extracted entities and relations. Once the semantic concept space, interactively adjusted and partially ranked by the user, is complete, the selected story path can be exported into the desired format for further processing. This functionality will be implemented in collaboration with our SME partners and tailored to their respective in-house systems.

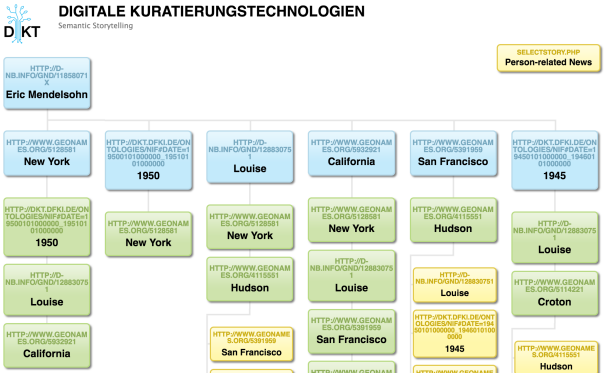


Figure 3: Filled in *News* story template relating to the root concept "Erich Mendelsohn"

4 Use Case and Experiments

The development of the DS3 system is work in progress. In the following we describe an experiment that relates to a typical future use case of the system. Since a key requirement of the whole Digital Curation Technologies project is adaptability to new and specialised domains, we made several experiments with the Mendelsohn Letters, a large set of letters written by Erich Mendelsohn, a well-known German architect.[3] With the current DS3 prototype, we can extract required information and fill templates by using DBPedia in combination with template-specific SPARQL queries. However, for smaller and more specialised domains such a (complete) ontology may not be available. For the Mendelsohn experiments we adapted our NER module to this domain and created semantic annotations for a random sample of 1,000 letters. We extracted the relation triples and evaluated if they are suitable for filling the selected templates. Given our currently still limited set of templates, the amount of information obtained was also limited. We were able to extract several relations from the data set, but only some were useful for filling slots in the templates due to the limited recall of the dependency-based relation extraction (see Section 3.3). In an attempt to extract additional relation candidates, we assumed entities to be related if they appear near each other. Of the window sizes we tried (within 5, 10 and 20 words of each other), we found that a size of 20 gave the best results. This is also what we used to generate the tree shown in Figure 3, where any entities that appear more than ten times together in the same window are shown. This rather coarse-grained approach of finding potentially related entities serves as an alternative, only to demonstrate what our current output looks like. Not in the least place because this proximity approach would only establish the subject and object of the relation triples we use and not the relation type itself, which is taken from the verb through the dependency parsing approach. Finding more informative relation triples using more general and more robust relation extraction approaches with a bigger coverage will be an important next step. Because the individual components we use in the platform (except for the temporal analyser) are typical off-the-shelf implementations with limited modifications, we do not provide F scores for these components. Instead, the focus is on combining existing technologies within a larger platform for Digital Curation Technologies, especially with regard to Semantic Story Telling. As a next step we will do an evaluation in which we will ask a group of knowledge workers to compare their workflow with and without using our tools. A key principle of the project is that the human expert is always in the loop. This means that the performance of individual components is secondary to the efficiency and usability of the services and platform as a whole and evaluation should be user-oriented.

[3] http://ema.smb.museum/en/home/

5 Summary and Future Work

We are developing a system that will support knowledge workers in the complex and time-consuming task of handling, evaluating, processing, sorting and processing of document collections – either data in motion coming in, among others, from online news wires, or highly specialised, self-contained document collections. The primary goal of the system is to enable journalists, editors, authors, i. e., curators of digital content to identify interesting story lines as efficiently as possible. The current prototype is able to fill manually selected story templates based on semantically processing a document collection through, for example, named entity recognition and relation extraction. Future work includes the implementation of additional semantic analysis modules (e. g., entity recognition with higher recall through classic IR methods such as TF/IDF, additional as well as template-specific relation extraction methods, exploiting ontologies to make better use of identified relations), more detailed template descriptions, additional templates and crosslingual capabilities through machine translation. Within the context of the project Digital Curation Technologies we plan to test the system in two newsrooms (newspaper, television station) and also in a digital agency that specialises on designing and curating online portals for cultural archives and heritage information.

Acknowledgments: The authors would like to thank the anonymous reviewers for their helpful comments and suggestions. The project Digitale Kuratierungstechnologien is supported by the German Federal Ministry of Education and Research, Unternehmen Region, Wachstumskern-Potenzial (No. 03WKP45). The project FREME has received funding from the EU's Horizon 2020 programme under grant agreement No. 644 771.

References

[Bourgonje *et al.*, 2016] P. Bourgonje, J. Moreno-Schneider, J. Nehring, G. Rehm, F. Sasaki, and A. Srivastava. Towards a Platform for Curation Technologies: Enriching Text Collections with a Semantic-Web Layer. In H. Sack, G. Rizzo, N. Steinmetz, D. Mladeni, S. Auer, and C. Lange, editors, *The Semantic Web: ESWC 2016 Satellite Events*, June 2016. In print.

[Lewis *et al.*, 2014] D. Lewis, A. Gómez-Pérez, S. Hellmann, and F. Sasaki. The role of linked data for content annotation and translation. In *Proc. of 2014 European Data Forum*, 2014.

[Mann and Thompson, 1988] W. C. Mann and S. A. Thompson. Rhetorical Structure Theory: Toward a Functional Theory of Text Organization. *Text*, 8:243–281, 1988.

[Manning *et al.*, 2014] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky. The Stanford CoreNLP natural language processing toolkit. In *ACL System Demonstrations*, pages 55–60, 2014.

[Meghini and Bartalesi, 2014] C. Meghini and V. Bartalesi. Steps towards Enhancing the User Experience in Accessing Digital Libraries. In *Human Interface and the Management of Information. Information and Knowledge in Applications and Services – 16th Int. HCI Conference*, pages 555–566, Heraklion, Greece, 2014.

[Orlikowski and Yates, 1994] W. J. Orlikowski and J. Yates. Genre Repertoire: The Structuring of Communicative Practices in Organizations. *Administrative Science Quarterly*, (39):541–574, 1994.

[Rehm and Sasaki, 2016] G. Rehm and F. Sasaki. Digital Curation Technologies. In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation (EAMT 2016)*, Riga, Latvia, May 2016. In print.

[Rehm, 2002] G. Rehm. Towards Automatic Web Genre Identification – A Corpus-Based Approach in the Domain of Academia by Example of the Academic's Personal Homepage. In *Proceedings of the 35th Hawaii International Conference on System Sciences (HICSS-35)*, Big Island, Hawaii, January 2002.

[Rehm, 2005] G. Rehm. *Hypertextsorten: Definition – Struktur – Klassifikation*. PhD thesis, Institut für Germanistik, Angewandte Sprachwissenschaft und Computerlinguistik, Justus-Liebig-Universität Gießen, 2005.

[Rumelhart, 1975] D. E. Rumelhart. Notes on a Schema for Stories. In Daniel G. Bobrow and Allan Collins, editors, *Representation and Understanding – Studies in Cognitive Science*, pages 211–236. Academic Press, New York, San Francisco, London, 1975.

[Sasaki *et al.*, 2015] F. Sasaki, T. Gornostay, M. Dojchinovski, M. Osella, E. Mannens, G. Stoitsis, P. Richie, T. Declerck, and K. Koidl. Introducing FREME: Deploying Linguistic Linked Data. In *Proc. of 4th Multilingual Semantic Web Workshop*, 2015.

[Y.-C. Li, 2012] W.-P. Su Y.-C. Li, A. Wee-Chung Liew. The Digital Museum: Challenges and Solution. *Information Science and Digital Content Technology*, pages 646–649, 2012.

Automatic Creation of Flexible Catchy Headlines

Lorenzo Gatti
FBK-irst
Trento, Italy
l.gatti@fbk.eu

Gözde Özbal
FBK-irst
Trento, Italy
gozbalde@gmail.com

Marco Guerini
FBK-irst
Trento, Italy
guerini@fbk.eu

Oliviero Stock
FBK-irst
Trento, Italy
stock@fbk.eu

Carlo Strapparava
FBK-irst
Trento, Italy
strappa@fbk.eu

Abstract

In this paper we present a creative system for producing news headlines based on well-known expressions. The algorithm is composed of several steps that identify keywords from a news article, select an appropriate well-known expression and modify it via word replacement or insertion to produce a novel headline, using state-of-the-art natural language processing and linguistic creativity techniques. A simple web-interface abstracts the technical details from users, and lets them focus on the task of producing creative headlines.

1 Introduction

In a web column on the New York Times with title "Headline Art"[1], Stanley Fischer speaks in admiration of famous attention-grabbing headlines and headlines that demand interpretive work of a kind usually associated with modern poetry. The value of catchy headlines has become more and more important as they can be considered as the first reader entry points, and in many cases a catchy headline is a fundamental prerequisite for the success of news. Several traditional newspapers have invested in acquiring brilliant creative headline producers to make sure they win the competition on people's attention.

This competition is becoming more pervasive with the advent of electronic news where the entry point to the newspaper can be any article (e.g., as seen on the news feed on social media sites), not just the "front page" of the newspaper. Therefore, creative titles are needed for each article in addition to front articles. The problem is that the feasibility and overall costs of this process, if based on creative humans, prevent this development. These considerations motivate the need of automatic, or at least semi-automatic production of headlines. Creative natural language processing is showing some promising results, not only in domains such as poetry [Toivanen *et al.*, 2012], novel metaphors [Veale, 2014] or humor production [Binsted and Ritchie, 1997; Stock and Strapparava, 2006], but has also displayed a potential in applied areas such as catchy advertisement production [Valitutti *et al.*, 2009].

A concept often exploited in creative language production is to give novel life to a known expression by adapting it to a new situation. For instance, the revised expression may evoke some of the news of the day, while keeping the original expression still perceivable, like in: "Naming Private Ryan"[2]. The effectiveness of this process is correlated with the aesthetic pleasure involved in the appreciation of the modification. The news of the day, which obviously cannot be predicted in advance, can be promoted through a creative tagline based on the parasitic use of an automatically selected linguistic expression (for instance a slogan, movie title or well known quote). To achieve that, the expression can be slightly modified into a novel one that evokes the news by still winking to its origin.

In this paper, we present a system called Heady-Lines, which automatically proposes catchy headlines for news appearing, for instance, on the online edition of the New York Times or the BBC. Heady-Lines uses a linguistically motivated framework that accounts for syntagmatic and paradigmatic aspects of language. It receives short descriptions of news articles as input and utilizes various NLP techniques to innovate existing well-known expressions by bringing in a new concept coming from evolving news. The revised expression aims to evoke the targeted news while keeping the original expression still perceivable.

We use semantic similarity metrics to pair a well-known expression with an appropriate news article. In addition, we use morpho-syntactic constraints and the dependency structure of the expression to modify it into a meaningful and grammatical headline. As for aesthetics and the involved cognitive aspects, our "ideological" reference is the so called Op-

[1] http://opinionator.blogs.nytimes.com/2009/04/19/headline-art

[2] http://www.mirror.co.uk/sport/football/news/mp-outed-manchester-united-star-3328307

timal Innovation Hypothesis [Giora, 2003]. This theory states that variations from a known text are highly appreciated, and almost invariably more pleasurable than an entirely new text. In particular, to be "optimally innovative" a text has to evoke a novel response, while still allowing for the recovery of a salient response (i.e. an expression that the reader is familiar with), from which it differs, so that both can be weighed against each other. To the best of our knowledge, Heady-Lines is the first attempt at blending well known expressions with recent news in a linguistically motivated framework that accounts for syntagmatic and paradigmatic aspects of language.

The system is paired with an easy-to-use interface that empowers users to start from one of these news articles and select a new creative headline among those proposed by Heady-Lines.

2 Related Work

Poetry generation systems face similar challenges to ours as they struggle to combine semantic, lexical and phonetic features in a unified framework. Greene et al. [2010] describe a model for poetry generation in which users can control meter and rhyme scheme. Toivanen et al. [2012] propose to generate novel poems by replacing words in existing poetry with morphologically compatible words that are semantically related to a target domain. Colton et al. [2012] present another data-driven approach to poetry generation based on simile transformation, where daily news influence the mood and theme of the poems. After presenting a series of web services for novel simile generation, divergent categorization, affective metaphor generation and expansion, Veale [2014] introduces *Stereotype*, a service for metaphor-rich poem generation. Given a topic as input, *Stereotype* combines the previous services to obtain a master metaphor, its elaborations, and proposition-level world knowledge. All these ingredients are then packaged by the service into a complete poem.

Recently, some attempt has been made to generate creative sentences for educational and advertising applications. Özbal et al. [2013] propose an extensible framework called BRAINSUP for the generation of creative sentences in which users are able to force several words to appear in the sentences. BRAINSUP makes heavy use of syntactic information to enforce well-formed sentences and to constraint the search for a solution, and provides an extensible framework in which various forms of linguistic creativity can easily be incorporated. An extension of this framework is used in a more recent study [Özbal *et al.*, 2014] to automate and evaluate the keyword method, which is a common technique to teach second language vocabulary.

As a notable study focusing on the modification of linguistic expressions, the system called Valentino [Guerini *et al.*, 2011] slants existing textual expressions to obtain more positively or negatively valenced versions by using WordNet [Miller, 1995] semantic relations and SentiWords [Gatti *et al.*, 2015b]. The slanting is carried out by modifying, adding or deleting single words from existing sentences. The modification is performed first based on the dependents from left to right and then possibly the head. Valentino is also used to spoof existing ads by exaggerating them, as described in [Gatti *et al.*, 2014], which focuses on creating a graphic rendition of each parodied ad.

Lexical substitution has also been commonly used by various studies focusing on humor generation. Stock and Strapparava [2006] generate acronyms based on lexical substitution via semantic field opposition, rhyme, rhythm and semantic relations provided by WordNet. The proposed model is limited to the generation of noun phrases. Valitutti et al. [2009] present an interactive system which generates humorous puns obtained by modifying familiar expressions with word substitution. The modification takes place considering the phonetic distance between the replaced and candidate words, and semantic constraints such as semantic similarity, domain opposition and affective polarity difference.

Finally, it is worth mentioning that many creative systems are based on the Conceptual Blending Theory [Fauconnier and Turner, 2008], a framework for mapping two concepts from different domains into a new "blended space", which inherits properties from both starting domains. In our work, however, we insert a new concept into an existing expression by replacing or inserting a word, without modeling the starting domains and finding their shared properties.

3 Heady-Lines

Heady-Lines is composed of four main modules that (*i*) retrieve the news of the day from the web, (*ii*) extract keywords from the news and expand them with relevant related concepts, (*iii*) pair the news with well-known expressions using state-of-the-art similarity metrics, (*iv*) generate a new headline by merging the well-known expression with a keyword coming from the news, satisfying the lexical and morpho-syntactic constraints enforced by the expression. Since Heady-Lines is meant to be used as an aid for copy editors, the interface hides these technical details and collaborates with the users in the creative task of generating a good headline. However, the system can also work in a fully-automatic mode, where just a news (or a feed of news) is given in input and the system presents what it thinks is the best headline.

In the remainder of this section we provide an overview of the process for creating a new headline. The process is the same both in fully-automatic and in interactive mode, but in the second case the user can intervene in some parts of the process to guide the system, correcting possible errors and exploring alternatives that the automatic mode would normally discard. For full details, please refer to [Gatti *et al.*, 2015a].

3.1 Selecting a news article

Users are presented a list of short descriptions (about 25 words) of the news of the day. Since the headlines generated with the Heady-Lines process are often humorous, and thus not appropriate for tragic events that often appear in the news, a slider on the top allows users to filter negative descriptions and focus only on positive news. In automatic mode, the system simply discards negative news to avoid creating instances of black humour.

From a technical point of view, the news are retrieved from the RSS feed of BBC News and the New York Times through

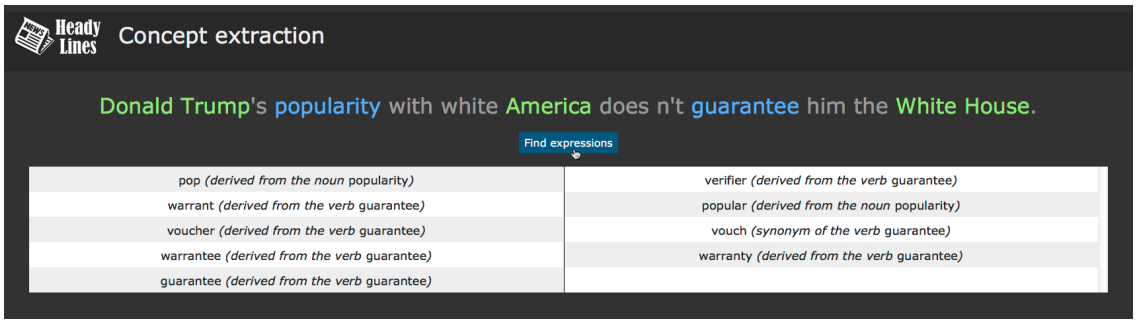


Figure 1: Keyword identification and expansion

its API. Each entry is composed of a headline, the short description of the article and other metadata, but only the description is used by the system. As they are downloaded, news are tokenized and PoS-tagged using Stanford CoreNLP [Manning *et al.*, 2014], which also provides the sentiment labels that we use for the filter.

An example to a description provided by our interface is "Ukraine has become "very volatile" since Prime Minister Arseny Yatseniuk resigned, the head of the Council of Europe said on Monday, calling for the swift formation of a new government and speedier progress on reforms." (from the NYT).

3.2 Selecting keywords

Once an interesting news event is selected from the list, its description is presented in a new page with its key concepts highlighted (Figure 1). In particular, the interface makes stop words and irrelevant words fade to grey, while the defining elements for that news are differently colored, depending on their category. At the moment we are differentiating among i) named entities, ii) important concepts for which we have some knowledge (i.e. they exist in WordNet), iii) important but "unrecognized" concepts. Users are also presented with an additional set of related keywords which are derived from the important words recognized in the sentence. When using the system in interactive mode, the user can remove any of the identified keywords or related concepts from the list, or force a new word or one in the description to be considered important.

We define the importance of each word as the number of times that a lemma appears in a news corpus (23,415 news documents from LDC GigaWord corpus [Parker *et al.*, 2011]), divided by the total number of headlines occurring in the corpus (i.e. the probability of the lemma). Lemmas under a certain threshold are considered as "key concepts". The "related keywords" that the users see are simply the synonyms and derivationally related forms of these previous lemmas obtained from WordNet. The named entities are detected with CoreNLP.

In the description of the previous example, the system identifies *volatile*, *Prime*, *Minister*, *Arseny Yatseniuk*, resigned, *Council of Europe*, *speedier* and *reforms* among the key terms. It expands this list by retrieving concepts such as the adjective *vacant* (derived from the adverb *resign*) and the noun *velocity* (from *speedy*).

3.3 Selecting a well-known expression

A list of well-known expressions is then presented to the user, with the expressions that are most related to the news appearing at the top (Figure 2). The list consists of approximately 100 elements including book, song and movie titles and common idioms (e.g. "live and let live"). They were chosen among the top-ranked in Billboard charts, online movie databases and lists of the most sold books of all time.

The relatedness is calculated using a skip-gram model [Mikolov *et al.*, 2013] trained on a lemmatized and PoS-tagged subset of the GigaWord corpus. To compare the news with each expression, we construct a vector representation of the former by summing the vectors of its keywords. Similarly, we build the vector representation of each expression based on its lemmas (after removing the stop words). Moreover, the expressions that do not reach a certain similarity threshold are discarded. This ensures at least a minimum degree of relatedness between the news and the well known expression. If no expression reaches the threshold, in fully-automatic mode the algorithm would stop and process the next headline in the input list, if any. In interactive mode, however, the user can manually select some of the expressions, forcing the system to consider them for the next steps.

Based on the same example, the most similar well-known expression is the song title "Wind of change".

3.4 Selecting the final headline

The list of the new potential headlines is then shown to the user (Figure 3), ranked from "best" to "worst" (i.e. the system ranks them by similarity and grammaticality, ensuring that the first headline is the one that maximizes these two metrics). In interactive mode the user can click on any of the sentences and see a final page with the headline, along with the starting description, to see how fit it is for the news. When working in automatic mode, however, Heady-Lines simply chooses the top headline.

The sentences are modified either by replacing an existing word, or by inserting a new one in the well-known expression. In both cases, the modifications take into account the lexical and syntactic constraints imposed by the original expression. To do this, we use a database of tuples that stores, for each relation in the dependency treebank of LDC GigaWord corpus, its occurrences with specific "governors" (heads) and

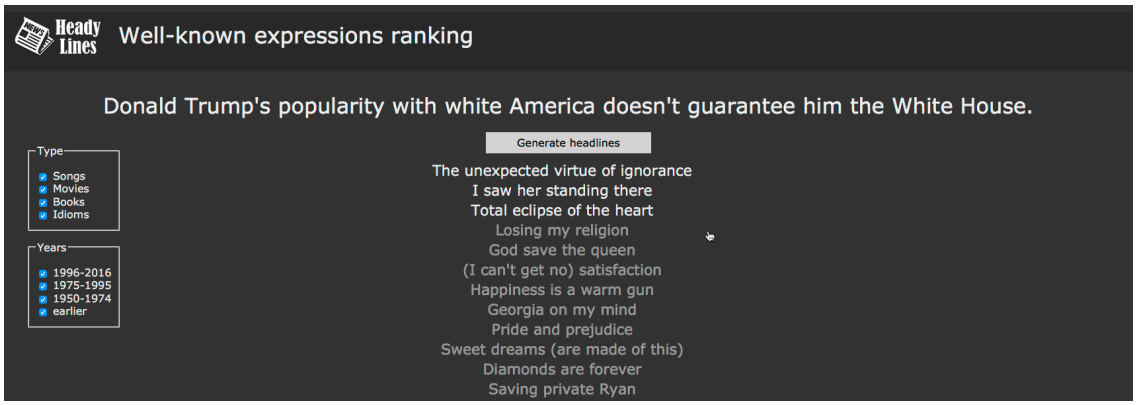


Figure 2: Selecting the well-known expressions

Figure 3: Selecting a final headline

"dependents" (modifiers), similarly to the approach of Özbal et al. [2013].

For the replacements, for each lemma w in an expression, we determine all the words d_i that are connected to w in a dependency relation r. Then, we calculate how likely each keyword k, coming from the news articles that passed the similarity filter, can replace a w of the same part-of-speech. This is done by considering how frequently k is in a r relation with all the d_i, in our reference corpus. We can then select the slot with the word w to be replaced, and the best keyword k for each news article, by maximizing this dependency likelihood. Finally, the morphology of the replaced word w is applied to k using MorphoPro [Pianta *et al.*, 2008] and the headline is generated.

For the insertions, an adjective (or adverb) k is inserted before the noun (verb) w that appears most often in an appropriate dependency relation with it (i.e. "amod" and "advmod" respectively).

For each expression we try to generate headlines with both replacements and insertions. Then, the one with the best dependency score between the two (i.e. the most "grammatical") is chosen, so that from each expression we produce at most one headline. Also in this case a threshold is enforced, so that headlines that do not reach a satisfactory level of grammaticality are removed. To rank the final output, the system sorts each modified sentence according to its mean rank with respect to similarity and dependency scores, thus balancing the scores of grammaticality and relatedness to the news. The lower the mean, the better the system considers the headline.

In our example, the system will choose "Wind of rapid change" as the best headline. More examples of the system output are shown in Table 1.

4 Conclusions

In this paper, we presented a system that utilizes various NLP techniques to generate creative headlines by modifying existing well-known expressions with concepts coming from the news. An initial evaluation [Gatti *et al.*, 2015a] confirmed the effectiveness of this system. We believe that this effectiveness is correlated with the aesthetic pleasure involved in the appreciation of the modification. In any case, the automation of this kind of creative process can be of great use in many fields such as journalism, advertising and applied arts.

As future work, we plan to improve the sorting mechanism with the addition of a memorability score for the modified expression, and add other modification strategies, in particular one based on phonetic properties of the replacement words.

Moreover, we would like to experiment with the concept of personalization, i.e. presenting different material to different groups of people, taking into account their interest or, age or any other information that is available. We think an interesting developement would be producing personalized headlines for attracting different groups of people; or even to go one step forward, introducing dynamic adaptivity, where some considerations about the state of an individual (such as the emotional state induced by a personal event) may ideally

Description	Ukraine has become "very volatile" [...], the head of the Council of Europe said on Monday, calling for [...] speedier progress on reforms.
Expression	Wind of change
Headline	Wind of *rapid* change
Description	The Obama administration is planning to issue a final rule designed to enhance the safety of offshore oil drilling equipment.
Expression	Bridge over troubled water
Headline	Bridge over troubled *oily* water
Description	Russia's defense ministry has rejected complaints by U.S. officials who claimed Russian attack planes buzzed dangerously close to a U.S. Navy destroyer [...]
Expression	The empire strikes back
Headline	The *Russian* empire strikes back
Description	Pyongyang drivers are feeling some pain at the pump as rising gas prices put a pinch on what has been major traffic growth [...]
Expression	House of the rising sun
Headline	House of the rising *prices*

Table 1: Output examples

be usable for flexible, electronic-based personal news production.

References

[Binsted and Ritchie, 1997] Kim Binsted and Graeme Ritchie. Computational rules for generating punning riddles. *Humor - International Journal of Humor Research*, 10(1):25–76, 1997.

[Colton *et al.*, 2012] Simon Colton, Jacob Goodwin, and Tony Veale. Full-FACE poetry generation. In *Proceedings of ICCC'12*, pages 95–102, 2012.

[Fauconnier and Turner, 2008] Gilles Fauconnier and Mark Turner. *The way we think: Conceptual blending and the mind's hidden complexities*. Basic Books, 2008.

[Gatti *et al.*, 2014] Lorenzo Gatti, Marco Guerini, Oliviero Stock, and Carlo Strapparava. Subvertiser: mocking ads through mobile phones. In *Proceedings of IUI'14*, pages 41–44, 2014.

[Gatti *et al.*, 2015a] Lorenzo Gatti, Gözde Özbal, Marco Guerini, Oliviero Stock, and Carlo Strapparava. Slogans are not forever: adapting linguistic expressions to the news. In *Proceedings of of IJCAI'15*, 2015.

[Gatti *et al.*, 2015b] Lorenzo Gatti, Marco Turchi, and Marco Guerini. Sentiwords: Deriving a high precision and high coverage lexicon for sentiment analysis. *IEEE Transactions on Affective Computing*, Preprints(1):1–11, 2015.

[Giora, 2003] Rachel Giora. *On Our Mind: Salience, Context and Figurative Language*. Oxford University Press, New York, 2003.

[Greene *et al.*, 2010] Erica Greene, Tugba Bodrumlu, and Kevin Knight. Automatic analysis of rhythmic poetry with applications to generation and translation. In *Proceedings of EMNLP'10*, pages 524–533, 2010.

[Guerini *et al.*, 2011] Marco Guerini, Carlo Strapparava, and Oliviero Stock. Slanting existing text with Valentino. In *Proceedings of IUI'11*, pages 439–440, 2011.

[Manning *et al.*, 2014] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of ACL'14*, pages 55–60, 2014.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS'13*, pages 3111–3119, 2013.

[Miller, 1995] George A. Miller. WordNet: A lexical database for English. *Communications of the ACM*, 38:39–41, 1995.

[Özbal *et al.*, 2013] Gözde Özbal, Daniele Pighin, and Carlo Strapparava. BRAINSUP: Brainstorming support for creative sentence generation. In *Proceedings of ACL'13*, pages 1446–1455, 2013.

[Özbal *et al.*, 2014] Gözde Özbal, Daniele Pighin, and Carlo Strapparava. Automation and evaluation of the keyword method for second language learning. In *Proceedings of ACL'14*, pages 352–357, 2014.

[Parker *et al.*, 2011] Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. English gigaword 5th ed. *Linguistic Data Consortium, LDC2011T07*, 2011.

[Pianta *et al.*, 2008] Emanuale Pianta, Christian Girardi, and Roberto Zanoli. The TextPro tool suite. In *Proceedings of LREC'08*, pages 2603–2607, 2008.

[Stock and Strapparava, 2006] Oliviero Stock and Carlo Strapparava. Laughing with HAHAcronym, a computational humor system. In *Proceedings of AAAI'06*, pages 1675–1678, 2006.

[Toivanen *et al.*, 2012] Jukka M. Toivanen, Hannu Toivonen, Alessandro Valitutti, and Oskar Gross. Corpus-based generation of content and form in poetry. In *Proceedings of ICCC'12*, pages 175–179, 2012.

[Valitutti *et al.*, 2009] Alessandro Valitutti, Carlo Strapparava, and Oliviero Stock. Graphlaugh: a tool for the interactive generation of humorous puns. In *Proceedings of ACII'09 Demo track*, pages 634–636, 2009.

[Veale, 2014] Tony Veale. A service-oriented architecture for metaphor processing. In *Proceedings of the Second Workshop on Metaphor in NLP*, pages 52–60, 2014.

Helping News Editors Write Better Headlines:
A Recommender to Improve the Keyword Contents & Shareability of News Headlines

Terrence Szymanski, Claudia Orellana-Rodriguez, Mark T. Keane
Insight Centre for Data Analytics &
School of Computer Science
University College Dublin
{terrence.szymanski,claudia.orellana,mark.keane}@insight-centre.org

Abstract

We present a software tool that employs state-of-the-art natural language processing (NLP) and machine learning techniques to help newspaper editors compose effective headlines for online publication. The system identifies the most salient keywords in a news article and ranks them based on both their overall popularity and their direct relevance to the article. The system also uses a supervised regression model to identify headlines that are likely to be widely shared on social media. The user interface is designed to simplify and speed the editor's decision process on the composition of the headline. As such, the tool provides an efficient way to combine the benefits of automated predictors of engagement and search-engine optimization (SEO) with human judgments of overall headline quality.

1 Introduction

The headline is an extremely important component of every news article that performs multiple functions: summarizing the story, attracting attention, and signaling the voice and style of the newspaper [Conboy, 2007]. In the online realm, headlines are expected to meet several new functions; for instance, to convey the article's contents in different online contexts or to optimize the article for search engine queries (i.e, SEO). Indeed, arguably, the headline is now more important than ever, as it becomes the only visible part of the article in microblog posts, social media feeds and listings on news-aggregation sites. These multiple requirements on the news headline have complicated the composition task facing news editors, as they attempt to ensure that each headline is crafted as perfectly as possible.

Prior NLP work in the area of news headlines has mostly focused on the task of automatic headline generation, cast as "very short summary generation" in the DUC tasks of the early 2000s; tasks that produced much of the research on the topic. The best-performing system in the 2004 DUC task worked by parsing the first sentence of the article and pruning it to the desired length [Zajic *et al.*, 2004], an approach that works by leveraging human intelligence: journalists generally compose news articles in the "inverted pyramid" style, which places the most important information in the lead paragraph [Conboy, 2007]. Other headline generation systems generally work by first using some metric to identify terms within the document that are likely to appear in the headline, and then constructing a headline containing these terms [Nenkova and McKeown, 2011].

This latter approach has much in common with the task of keyword selection for SEO, which first caught the attention of major newspapers at least ten years ago [Lohr, 2006], and continues to be a much-discussed issue today [Sullivan, 2015]. While even long-established, traditional news publications have begun to move away from classical forms of headlines towards more direct, keyword-laden headlines, many copy editors would still prefer to write clever, witty headlines [Wheeler, 2011], and readers of the news seem to value creativity in headlines over clarity or informativeness [Ifantidou, 2009]. Therefore, one of the key considerations in the design of our system was to balance the mechanical act of filling a headline with informative, relevant keywords, against the creative act of writing headlines that appeal to human interests and emotions.

We expect that the most interesting and emotional stories are likely to be more popular with readers than the "average" story. Analysis of reader behavior has shown that there is no correlation between how much an article is shared on social media and how much of the article is read by an average user [Haile, 2014]; a fact that could be taken as evidence supporting the widely-held view that people share articles online that they have not fully read themselves [Manjoo, 2013]. In this case, the headline—which people presumably read even if they don't read the full text—may be an important factor in determining the "shareability" of a news article; an idea that is another key motivation behind the design of our system.

The tool presented here is designed to facilitate the decision-making process facing a news editor in composing a headline. The software employs state-of-the-art NLP and machine learning techniques to make its recommendations, but it is not designed to automatically generate headlines or to make decisions about a headline's goodness on its own.

In the sections below, we present the design and behavior of the tool before discussing the internal workings of the system. We conclude with an assessment of the current state of the project, including some preliminary evaluation results and a discussion of areas for improvement.

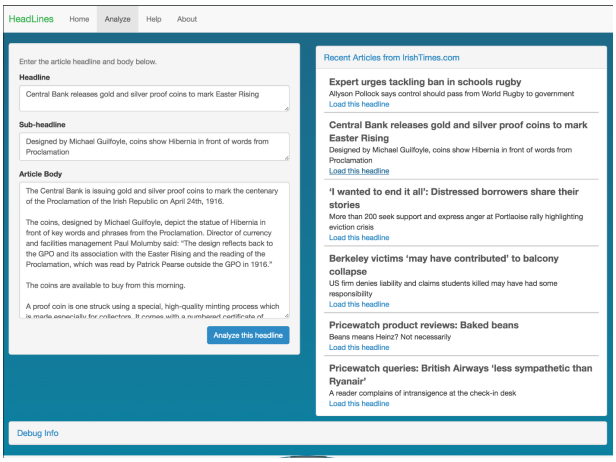


Figure 1: Screenshot of the tool in input mode, with input text areas on the left and a live feed of articles on the right.

2 Design and Behavior

From a user-interface perspective, the software has two modes of operation: input mode and analysis mode. The input mode (illustrated in Figure 1) facilitates the entry of a news article and its corresponding headline and sub-headline, which may either be entered manually or selected from a feed of recent articles. In practice, this feed would be integrated into the newspaper's workflow so that an editor could review all new articles with the software prior to publication.

After the editor-user selects an article, the system switches to the analysis mode, showing the results of the automated analysis (illustrated in Figure 2). This mode is designed to allow the user to quickly assess the strengths and weaknesses of the headline and decide whether any changes should be made to improve it. The five most highly-ranked keywords from the article are listed on the right side of the screen sorted by *weight*, a metric combining the keyword's *frequency* in the article and its *SEO score*, which respectively capture the keyword's local relevance to the article itself as well as its global prominence among news stories in general. (See section 3.1 below for details on these measures.)

The keywords are color-coded to distinguish keywords which already appear in the headline (green) from those which do not appear in the headline (red), and size-coded according to their weight. Thus, any large, red keywords are those which an editor should consider adding to the headline. In the example in Figure 2, the top three recommended keywords are already present in the headline; the two remaining recommendations, "Irish Republic" and "GPO", are both sensible suggestions for the article.

In addition to the keyword recommendations, the system scores each headline for its "shareability" on two social media platforms: Twitter and Facebook; if the shareability score on either platform exceeds a threshold value, then an alert is displayed to the user. In the example in Figure 2, the article has exceeded the Facebook threshold but not the Twitter threshold, so only one of the two alerts is displayed. The newspaper's editor in charge of social media can use this information when deciding which stories should be posted and promoted on social media sites. The threshold is set to a relatively conservative value, so that most articles will not produce alerts, and only the most promising headlines will come to the editor's attention.

Ultimately, it is up to the editor to decide what action, if any, to take based on the information presented by the software. The editor has the leeway to add keywords in the headline in creative ways that fit the style of the story and the news organization, and she can also flexibly deal with any errors that may be produced by the keyword recommendation system, rather than blindly following its advice.

3 Implementation

The system consists of three major components: a user-interface front-end, a text analysis back-end, and a web server that mediates communication between the two. The user interface is implemented with HTML and Javascript and accessed via a web browser; its behavior is described and illustrated in the previous section. The web server is implemented in Python (based on the *Flask* framework), which allows easy integration with the text analytic back-end, which is also mainly implemented in Python. We use the *sklearn* module for regression and Stanford's *CoreNLP* Java suite for NLP [Manning *et al.*, 2014]. The entire system is deployed on a web server and accessed by the client's web browser.

The back-end consists of two components—keyword analysis and shareability analysis—which operate independently of one another and are discussed in detail below.

3.1 Keyword Analysis

The role of keyword analysis is to identify terms in the article body that are good candidates for inclusion in the headline. We believe that headlines containing informative and popular keywords can be both more appealing to readers and more prominent in users' search results and on news aggregator websites.

Processing of an input article begins with tokenization and named-entity recognition using *CoreNLP*, which identifies all entities (e.g. people, organizations, locations) in the article. Next, any known keywords appearing in the text are identified, by using a database of 90k keywords and their frequencies from Irish news articles in recent years, which we populated with data provided to us by two other Irish news-related projects [Shi *et al.*, 2014; Bordea *et al.*, 2013]. This process results in a list of entities, which may be unique to the given article, and a list of keywords, which are known to have been encountered in previous news articles. These keywords and named entities are linked using a simple, rule-based approach that resolves pairs like "Enda Kenny" and "(Mr.) Kenny", yielding a single list of resolved keywords, along with a list of all positions in the text where each keyword appears.

Our keyword ranking system aims to capture the intuition that salient keywords should ideally be both *locally* prominent (i.e. appearing frequently in the given news article) and *globally* popular (i.e. appearing frequently in articles other then the current one). Thus, we calculate the weight w of

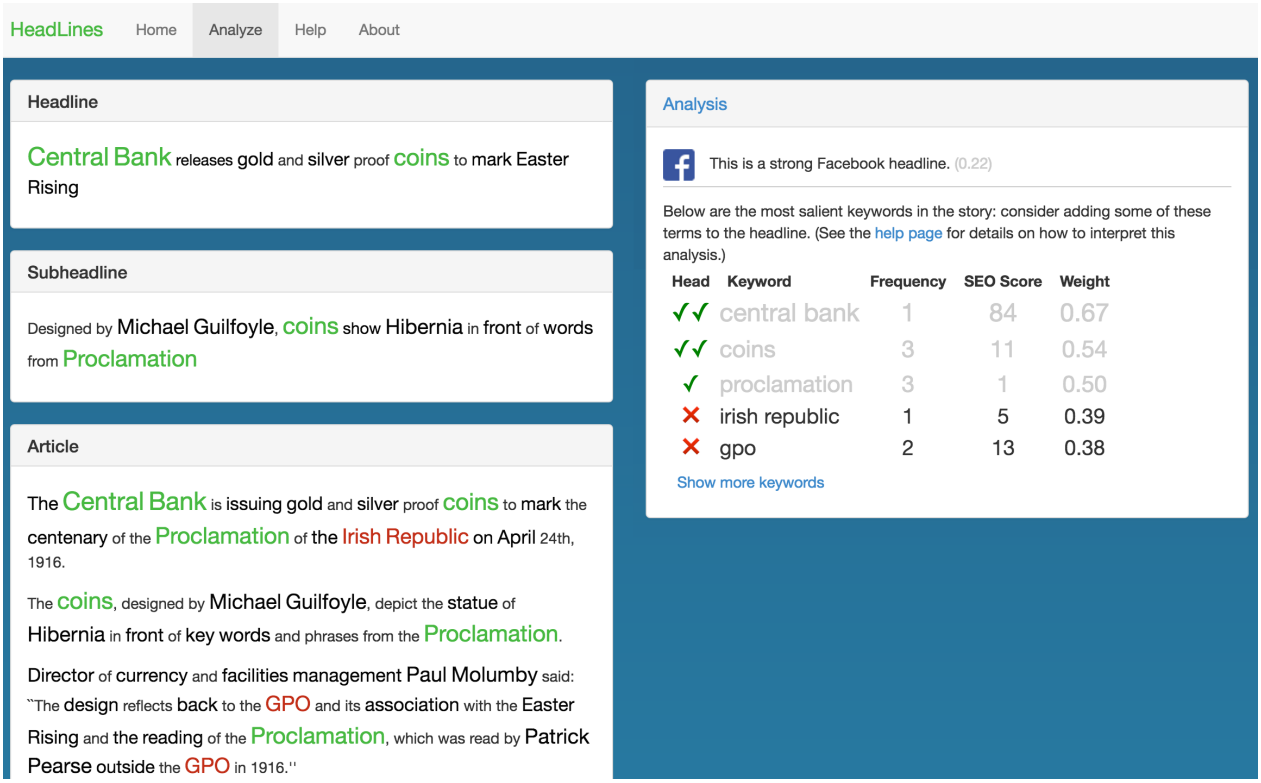


Figure 2: Screenshot of the tool in analysis mode. In this example, three of the top five keywords (highlighted in green) are already in the headline; the remaining two (in red) are recommendations that the user may consider adding to the headline.

each keyword k in the document d as the weighted sum of its local weight w_{local} and its global weight w_{global}:

$$w(k,d) = \lambda\, w_{local}(k,d) + (1-\lambda)\, w_{global}(k)$$

The local weight is calculated as the normalized within-document frequency of the keyword, so that the most frequent keyword in the document gets a w_{local} of 1. The global weight is calculated in a similar way, using the across-document frequencies from the keyword database and applying a nonlinear (log) transformation to compensate for the highly skewed distribution of these frequencies (note that it is possible for a keyword to have a zero global weight if it does not appear in our database; this is common for named entities in the article which have not been mentioned in the news before). The relative contributions of the local and global weights are balanced with the λ parameter.

This formula was chosen as the simplest method (a linear combination) of combining the two factors. It is similar to a *tf-idf* score in that it combines both term frequency and document frequency, but it is critically different in that it rewards, rather than penalizes, terms that occur in many documents. This is a good thing because we believe that terms which may be very common (e.g. the names of well-known politicians or celebrities) can be good headline terms, and also because our method of selecting terms (via a closed set of keywords and automatic named entity detection) generally avoids selecting words which may be high-frequency but low-quality (like stopwords).

We manually set the value of λ to achieve rankings which we subjectively deemed to be suitable.[1] This manual parameter setting allowed us to deploy our system quickly with acceptable performance, but a better option would be to learn these parameters automatically. To do so would require a dataset containing news articles, their headlines, and either some measure of the quality of the headline or an assurance that the headlines in the data set are "good", in order to guarantee that the parameters are set based on "good" headline examples. This type of data was not available to us when the system was under development.

This method ultimately assigns a weight to each keyword between 0 and 1.0, which determines its ranking in the analysis output (Figure 2). In the user interface, the weight is displayed in a table alongside the keyword's "frequency" and "SEO Score", which we consider to be more user-friendly

[1] We found that a value of 0.6 (i.e. slightly favoring local frequency over global frequency) worked well for our data, but this value changed depending on which keyword list we used. Ultimately, we combined both keyword lists, which introduced a large number of noisy terms. To suppress these noisy terms, we added an additional term to boost the score of keywords which were identified as named entities in the article (up to 0.2 of the overall weight).

than w_{local} and w_{global} themselves (the frequency is exactly the number of times the term appears in the article, and the SEO score is just w_{global} scaled to the familiar scale of 0 to 100).

3.2 Shareability Analysis

The role of shareability analysis is to identify headlines that are likely to be shared on social media. With the rise of social media as dissemination channels for the news, headlines now need to be both informative and "shareable"; that is, the headline somehow needs to attract people to post, share, and engage with the article on social media, in order to reach a large online audience.

According to the Reuters Institute Digital News Report [Newman *et al.*, 2016], Facebook and Twitter generate 54% of the visits to online news sites, suggesting that direct visits to the home pages of news providers are being supplanted by social media mediated access. However, Facebook and Twitter are known to have quite different audiences and engage users in different ways [Kirk *et al.*, 2015]. Users on Twitter generally actively search news and their consumption varies across news categories [Orellana-Rodriguez *et al.*, 2016], whereas on Facebook, news tends to be just encountered by sharing amongst friends. Therefore, in our system, we model the two social networks separately.

Using the Twitter streaming API we collected over 700k tweets and retweets posted by each one of 200 media outlets and journalist accounts for two time periods in 2013 and 2014, for a duration of 71 and 50 days, respectively. From the collected tweets we extracted all the URLs and used the Facebook and Twitter APIs to collect the number of times each URL was shared on Facebook or posted on Twitter. Because these posts were made by journalists, the links in the tweets are mainly to news articles, from which we extracted headlines. This step yielded a data set of 55k headlines with corresponding counts of social shares for each one.

We used a regression analysis to estimate the relationship between features of the headlines and the target variable of number of shares. Each headline in our collection is represented as a vector consisting of eight features covering three main aspects of the headline's content: the sentiment polarity (as computed by the *TextBlob* Python package), the presence of named entities, and the length in words. The complete list of features is presented in Table 1.

We used Regularized Linear Regression (RLR), Random Forest (RF) and Gradient Boosting Trees (GBT) as our methods for regression and used the metric Mean Squared Error (MSE) to assess their performance. We split our headlines set into 44k (80%) for training and the remaining 11k (20%) for testing. We train two different regression models, one for Facebook and one for Twitter. RF and GBT performed better than the RLR models. Between RF and GBT models, GBT performed slightly better than RF, although no significant difference was observed. On the basis of these results we use GBT as our method for regression. GBT have shown to outperform other models in classification and regression tasks and have been used successfully for audience engagement prediction [Diaz-Aviles *et al.*, 2014]. We observe that the models for Twitter and Facebook behave differently: comparing the values of the MSE for both models, predictions for shareable headlines on Facebook present an MSE of 41.8, while for Twitter the error is slightly smaller, 37.6.

Feature	Description
neutral	# of neutral sentiment words
positive	# of positive sentiment words
negative	# of negative sentiment words
organizations	# of ORGANIZATION entities
persons	# of PERSON entities
places	# of LOCATION entities
day	(T/F) headline contains the name of a day
length	total # of words in the headline

Table 1: Headline features used for regression. The first six features are normalized by the length of the headline.

Once the GBT models are trained, we store them and incorporate them into the system's pipeline. Every inputted headline receives two shareability scores, one for each social media site; however, in order to avoid triggering too many notifications to the journalist or news editor, the system only shows a result if the score is equal or larger than a manually-defined threshold of 3.7 and 1.7 for Facebook and Twitter, respectively, which correspond to the median number of shares (on each platform) received by the headlines in our collection.

4 Evaluation

The tool was developed in collaboration with The Irish Times, and several professional editors have tested its usability. The feedback from these sessions has been positive and has informed several design features. In particular, the color-coding and font-size features of the interface have been noted for their usability. On the basis of this success, we are now looking at integration into editors' daily workflow, to allow more usability data to be gathered.

Current tests of the system have identified some potential areas for improvement. The keyword system commonly fails to recognize when pairs of equivalent but non-identical keywords have the same referent; for example *Taioseach* and *Enda Kenny*, or *GPO* and *General Post Office*. While editors easily recognize this duplication, this error affects frequency counts, which in turn affect keyword rankings. This type of co-reference resolution is an open question in NLP research, with typical solutions relying on a rule-based or gazette-based approach to fix commonly-occurring cases.

The system could also be improved by moving from a static keyword database to a dynamic, real-time database. We were fortunate to be able to bootstrap our system with the keyword sources discussed in section 3.1; however neither of these sources were created with this specific use-case in mind, and the static nature of these lists means that the keyword database will become outdated over time. Updating the keyword frequency counts on a rolling basis is an easy first step; but a more sophisticated approach is probably required, where new entities are added to the database over time, and more recent articles are given a greater weight than older articles. Because our system already identifies named entities in

news articles, these entities could be added as new keywords in our database as they are encountered.

We are also evaluating the impact of using the tool on SEO, based on determining whether it improves article rankings in news aggregators and search engines. While the lack of click-through from Google News has led some to question its effectiveness at driving traffic to news sites [Wauters, 2010], for *The Irish Times'* website, Google News is a major source of referrals. An analysis of 30k *Irish Times* articles (from 1/10/15 to 31/3/2016) has shown that articles listed on the Google News (Irish edition) front pages received significantly ($p < 0.01$) more page views than unlisted articles; with Google News listed articles receiving almost twice as many views ($n = 11,125$, $\mu = 1665.5$ views per article) as unlisted articles ($n = 19,339$, $\mu = 892.4$ views per article). Google News' ranking algorithm is not publicly known, so the exact factors leading to this correlation are opaque; however, for practical purposes, if our keyword recommender leads to greater visibility on Google News, then we know it should increase readership.

Finally, the quality of our keyword recommendations can, in part, be assessed by noting whether the system's top-recommended keywords are already present in the original headline written for the article, as it shows that the system corresponds to human judgments (n.b., the keyword analysis only uses the article body, not the headline, as input). We processed a sample of roughly 3,000 Irish Times headlines with our system, and found that a majority of these (64%) contained two or more of the top five keywords recommended by our system (in either the headline or the subheadline), and a large majority (88%) contained at least one of the recommended keywords. We take this as evidence that the keywords recommended by our system generally correspond with the types of keywords that a human editor would normally include in the headline.

5 Conclusion

In this paper, we have presented a system for recommending keywords for inclusion in newspaper headlines and for identifying headlines with high potential shareability on social media. The system identifies plausible keywords that are both relevant to the given news article and popular overall in past news articles, in an effort to maximize both the reader interest and the SEO aspect of the headline. In addition, the system identifies headlines that are likely to receive above-average engagement on social media, allowing editors to effectively target their social media strategy. We believe that this tool can be a helpful component in modern, online-oriented newsrooms.

6 Acknowledgments

The authors would like to thank *The Irish Times* for their funding and help on this project. This work is supported by Science Foundation Ireland through the Insight Centre for Data Analytics under grant number SFI/12/RC/2289.

References

[Bordea *et al.*, 2013] G. Bordea, P. Buitelaar, and T. Polajnar. Domain-independent term extraction through domain modelling. In *Proceedings of TIA*, 2013.

[Conboy, 2007] M. Conboy. *The Language of the News*. Routledge, 2007.

[Diaz-Aviles *et al.*, 2014] E. Diaz-Aviles, H. T. Lam, F. Pinelli, S. Braghin, Y. Gkoufas, M. Berlingerio, and F. Calabrese. Predicting user engagement in Twitter with collaborative ranking. In *Proceedings of the 2014 RecSys Challenge*, 2014.

[Haile, 2014] T. Haile. What you think you know about the web is wrong. *Time Magazine*, 2014.

[Ifantidou, 2009] E. Ifantidou. Newspaper headlines and relevance: Ad hoc concepts in ad hoc contexts. *Journal of Pragmatics*, 41(4):699–720, 2009.

[Kirk *et al.*, 2015] N. Kirk, J. Suiter, and P. McNamara. *Reuters Institute Digital News Report (Ireland)*. 2015.

[Lohr, 2006] S. Lohr. This boring headline is written for Google. *The New York Times*, April 9 2006.

[Manjoo, 2013] F. Manjoo. You won't finish this article. *Slate Magazine*, June 6 2013.

[Manning *et al.*, 2014] C. D. Manning, M. Surdeanu, et al. The Stanford CoreNLP natural language processing toolkit. In *ACL System Demonstrations*, 2014.

[Nenkova and McKeown, 2011] A. Nenkova and K. McKeown. Automatic summarization. *Foundations and Trends in Information Retrieval*, 5(2–3), 2011.

[Newman *et al.*, 2016] N. Newman, R. Fletcher, D. A. L. Levy, and R. K. Nielsen. *Reuters Institute Digital News Report*. 2016.

[Orellana-Rodriguez *et al.*, 2016] C. Orellana-Rodriguez, D. Greene, and M. T. Keane. Spreading the news: How can journalists gain more engagement for their tweets? In *Proceedings of the 8th ACM Conference on Web Science*, 2016.

[Shi *et al.*, 2014] B. Shi, G. Ifrim, and N. Hurley. Be in the know: Connecting news articles to relevant Twitter conversations. In *Proceedings of ECML*, 2014.

[Sullivan, 2015] M. Sullivan. Hey, Google! Check out this column on headlines. *The New York Times*, April 18 2015.

[Wauters, 2010] R. Wauters. Report: 44% of Google News visitors scan headlines, don't click through. *TechCrunch*, January 19 2010.

[Wheeler, 2011] D. R. Wheeler. 'Google doesn't laugh': Saving witty headlines in the age of SEO. *The Atlantic*, May 11 2011.

[Zajic *et al.*, 2004] D. Zajic, B. J. Dorr, and R. Schwartz. BBN/UMD at DUC-2004: Topiary. In *Proceedings of DUC*, 2004.

Getting to know large newsflows: Automatically induced information structures as keyphrases for news content analysis

Samia Touileb
University of Bergen,
samia.touileb@uib.no

Katherine Duarte
University of Bergen,
katherine.duarte@uib.no

Abstract

We propose an approach to analyze large news corpora. We use a grammar induction algorithm that identifies salient information structures in a news corpus, in order to extract keyphrases constituting summaries. The information structures are representations of semantic content and represent the most salient information in a corpus. We use a manually generated codebook to evaluate the induced structures. Our method is applied to a Norwegian news corpus of 11.000 online and print news articles mentioning the keyword climate change, that reflects diverse topics and different points of view. Results suggest that automatically induced structures can be used to clarify the content of a large corpus by providing an overview characterizing it.

1 Introduction

Media scholars interested in journalism studies, as well as journalists, are today facing the huge amount of online data. They need tools developed to fit their needs and requirements, in order to preprocess, organize and quickly have an overview of the content of the news. The main purpose of the tools would be to help them to easily, and in a straightforward way, understand the content of thousands of pieces of news that are sometimes simultaneously released online.

They therefore need advanced text mining methods to search for and extract the most important news content. Having access to overviews of the content can help them grasp the whole story without the need to read massive amounts of news articles. The methods should uncover similarities in the language use by presenting overviews of the information at the sentence and the discussion level.

We aim to present an unsupervised method that will enable media scholars and journalists, and perhaps the reader, to easily capture and grasp how main ideas or concepts are discussed in large corpora. With this method, we bring something new to the table, an unsupervised approach that can provide context and language use patterns, in order to build up and get a sense of the whole story without having to search and read trough thousands of texts.

We present information structures that we believe represent sets of keyphrases summarizing content of news, and that can lead researchers' investigations. We use a grammar induction algorithm to induce information structures from news corpora. We consider these information structures as keyphrases representing overviews of the most salient information present in a corpus. We evaluate the automatically induced structures using a codebook manually developed, independently of this work, by media scholars for coding news articles dealing with the same issue.

In the following we motivate our work and use of a grammar induction algorithm (Section 2). In Section 3 we describe our method and the corpus in use. Section 4 presents some results and an evaluation of our method. In closing, Section 5 provides some conclusions to summarize our work and a discussion of potential future works.

2 Background

Many scholars have submerged into the field of climate change communication. Most media researchers have focused on traditional human coding of large corpora, and have mostly looked at climate change as a media attention issue (single case studies [Boykoff and Boykoff, 2007; Shaw, 2013; Liu *et al.*, 2011; Carvalho and Burgess, 2005] and cross-country studies [Boykoff *et al.*, 2015; Schmidt *et al.*, 2013; Eide and Kunelius, 2012]).

When performing human content analysis, there is a need for a fully explicated document: a codebook. The codebook should stand alone as a protocol for content analysis of messages [Neuendorf, 2002]. In an ideal world, the goal behind the codebook is to make a working tool as complete and as unambiguous as to almost eliminate the individual differences among coders [Neuendorf, 2002]. There are several approaches to generate a codebook: One can either write a comprehensive codebook with detailed descriptions of each category, variable and value. Another way to do it, is to have a rather slim codebook, and spend time to train the coders. Krippendorff [2004] defines content analysis as a research technique for making replicable and valid inferences from texts to the context of their use.

There are major differences between human coding and computer coding. Neuendorf [2002] describes human versus computer coding as human coding using people as coders, while computer coding is an automated approach to arrange variables according to a desired output formulated to the computer. The computers' capability of rapidly processing large

amounts of data is a main factor in content analysis. Furthermore, computers are useful because of their ability to process textual data reliably [Krippendorff, 2004]. Karlsson and Sjøvaag [2015] suggest that analysis of online media data cannot follow the same procedures as established content analysis, developed for analogue media formats. They argue that the established content analysis approach is insufficient to cope with the ever-changing scope of digital media and digital journalism, where time and space is constantly changing.

Automated methods can be used in order to explore, model and analyze the diverse contents of corpora. Most automated text analysis techniques used as part of social science methods follow a bag-of-words approach [Grimmer and Stewart, 2013]. Bag-of-words models do not reflect the structural properties of the language and ignore the word order. They therefore only capture the general "aboutness" of texts, but do little to uncover what is actually said about the various key concepts. The bag-of-words models are nevertheless useful, and have proved their efficiencies (see [Grimmer and Stewart, 2013]). However, there are regularities in the language, and words do not appear arbitrarily next to each other, but in a given position relative to other words. These regularities can be exploited to induce the most frequent word sequences within a corpus.

The distributional structure of a language has as results the use of a restricted number of words relatively to a certain term, especially in domain-specific corpora [Harris, 1954]. Analyzing only the surface form of a language can uncover its distributional structure [Harris, 1954]. Lamb [1961], attempting to automate Harris' idea, introduced the concept of grouping words into a sequence of horizontal elements (H-groups) and vertical elements (V-groups). Consider the example sentences "*Climate change is a reality*", "*Climate change is a hoax*", "*Climate change is a lie*", "*Global warming is a reality*", "*Global warming is a hoax*", "*Global warming is a lie*". The H-group here is the word sequence "is a", the V-groups are (Climate change, Global warming) and (reality, hoax, lie). Harris' insights have also become the foundation of some of the work in the field of grammar inference (for a review see [D'Ulizia *et al.*, 2011]). One example is the grammar induction algorithm ADIOS (Automatic DIstilation of Structure - [Solan *et al.*, 2005]). ADIOS has been modified for text mining purposes [Salway and Touileb, 2014], which we use in this work; and will refer to it hereafter by modADIOS.

Automatic keyphrase extraction is defined as being the "automatic selection of important and topical phrases from the body of a document" [Turney, 2000]. Keyphrase extraction methods extract a set of descriptive phrases from a given corpus, and have proven their potential and have been used for various Natural Language Processing (NLP) purposes (see [Hasan and Ng, 2014] for a review).

Our approach to extract keyphrases is different from the methods used in the literature; here we use a grammar induction algorithm to induce information structures, which contain (in their different forms i.e. H-groups and V-groups) a valid set of keyphrases. Another resemblance is the fact that the structures, as shown in [Touileb and Salway, 2014], contain the main concepts discussed in a corpus.

3 Approach

modADIOS induces structures representing the most salient information present in a corpus. The structures are induced from unannotated corpora based on statistical criteria that will induce patterns of language use around predetermined key terms of interest. In what follows, we describe the corpus at hand (Section 3.1). Then we introduce in Section 3.2 the codebook used during the evaluation of our method. Finally, we present our method in Section 3.3.

3.1 Corpus

We use a corpus of Norwegian news articles gathered from a media surveillance tool[1] with the search string "*klimaendring**" (climate change), where all possible forms of the term will be retrieved. We searched for the keyword in 168 print media newspapers (national, regional and local newspapers) and 185 online news media outlets.

The corpus is constituted of a collection of 11.000 news articles mentioning the keyword "*klimaendring**", and consists of 1000 articles from each year between 2006 and 2015 (from October to December), and 1000 articles between January and April 2016. We then focused on the 19.186 sentences containing our keyword "*klimaendring**".

3.2 Codebook

We use a codebook that has been subject to a traditional manual coding situation with rigorous revisions. The codebook has been developed and used in a transnational media network project [Eide and Kunelius, 2012]. The codebook comprises ten categories:

1. *Media: name of the newspaper*.
2. *Date*.
3. *Picture/illustration*: Stating the number of pictures, cartoons and visuals in the article.
4. *Story themes*: What is the story mainly about? The main story can either be (i) climate change, with no mention of extreme weather, (ii) climate change, extreme weather is mentioned, (iii) extreme weather, no mention of climate change, (iv) extreme weather, with mention of climate change, (v) story about something else, climate change and extreme weather are mentioned as secondary themes.
5. *Story size*: In number of words
6. *Genre: Reporting genres*: Can either be news, reportage, interviews, portraits, editorial, and so on.
7. *All the quoted voices*: National political system, inter- or transnational political system, NGOs and individual citizens, business actors, scientists and other experts or media/journalists.
8. *Number of voices*: Number of unique voices in the article.
9. *Main theme/Consequences*: Insurance/compensation, adaptation, mitigation, transportation, security, responsibility or lack of action, citizen responsibility, climate politics/summits/reports, trade union or other union, research (either climate research or other), technology, and other.

[1] Atekst: www.retriever-info.com/en/category/news-archive/

10. *Extreme weather and natural disasters*: Floods. heavy rain, avalanche/landslide, heat wave, cold wave, droughts, melting ice sheets, rising temperatures, storm/ hurricane, ocean currents, other extreme weather events related to wind and ocean, sea-level rise.

In this paper we do not address all of these categories. For example, a machine has to be trained to identify the content of both categories *6* and *8*, which require techniques beyond the scope of this paper. Category *3* cannot be identified since the corpus in use is a collection of texts, this means that any kind of pictures or illustrations are excluded. Categories *1*, *2* and *5* are categories representing metadata that can easily be retrieved; we therefore do not address them when analyzing the induced structures.

3.3 Method

In order to induce information structures from texts we use modADIOS [Salway and Touileb, 2014]). ADIOS [Solan *et al.*, 2005] is an unsupervised algorithm that discovers hierarchical structures in sequential data. It identifies the most significant patterns (similar to H-groups) and equivalence classes (similar to V-groups) within the context of patterns, using statistical information. The unannotated texts are presented as a set of sentences. In each iteration, the most significant pattern is identified with a statistical criterion that favors frequent sequences that occur in a variety of contexts. Then, the algorithm looks for possible equivalence classes within the context of the pattern. At the end of the iteration, the new pattern and equivalence class become vocabulary items and can become part of further to be induced structures, and hence hierarchical structures are formed.

modADIOS presents the structures in the form of regular expressions. Recall the previous example of climate change sentences in Section 2, the algorithm will induce information structures of the form "*((climate change|global warming) is a (reality|hoax|lie))*"; where the symbol "|" represent "or". The information structure can thus be read as: *climate change is a reality* **or** *climate change is a hoax* **or** *climate change is a lie* **or** *global warming is a reality* **or** *global warming is a hoax* **or** *global warming is a lie*.

The input is presented to modADIOS in the form of increasingly large snippets around key terms of interest [Salway and Touileb, 2014]. Focusing on snippets around a predefined key term lead the algorithm to only induce the structures present around it. The algorithm starts running on snippets with a small window of words around the key term (0 to 3 words on both sides of the key term, "*klimaendring**" in our case) and the window increases after a predetermined amount of iterations (the biggest snippet size contains from 10 to 12 words) [Salway and Touileb, 2014]. When the structures are induced, each structure is substituted with a unique ID in the text, and the algorithm proceeds running on the next snippet size. This in order to force the algorithm to find more patterning around the key terms and the previously induced structures [Salway and Touileb, 2014].

4 Results

We describe in this section some of the induced structures (4.1), we then present an evaluation of the method (4.2).

4.1 Information structures as keyphrases

We ran modADIOS on our news corpus: this resulted in a set of 359 structures around the keyword "*klimaendring**". Table 1 shows a small selection of these induced structures. The structures represent keyphrases providing different information about the content of the corpus. The structures are presented in Norwegian with their English translations bellow in italics.

Structures 1, 2, 6, and 8 show different ways and perspectives of discussing how climate change should be tackled and fought. Structures 3, 4, 7, 10, 13 and 14 all examine what climate change actually is and what caused it, and what are its effects and consequences.

Structure 5 mentions the different skepticism existing around the climate change issues. Structure 8 and 11 show the United Nations Conferences of Parties, where science and politics in regards to climate change are discussed. Structure 9 illustrates the current promises to reduce emissions, which are too little ambitious to actually prevent and stop the effects of climate change. Structure 12 shows a new kind of refugees, those fleeing from their homes due to climate change threats. Structure 15 is a structure representing the discussions around climate adaptations.

4.2 Evaluation

Evaluating a keyphrase extraction method, typically involves creating a mapping (an exact match) between the keyphrases in a gold standard and those produced by the method; then scoring the output using evaluation metrics such as precision, recall, and F-score. In this work, we do not do a traditional keyphrase extraction, since we use a grammar induction algorithm to induce information structures representing keyphrases; we therefore do not evaluate our method in a traditional way.

The set of automatically induced structures, identified as keyphrases, is evaluated by the extent to which they map to the categories of a codebook developed manually on a smaller corpus dealing with the same issue. Since the codebook has carefully been created by media scholars, we believe that it encompasses all the important information regarding the discussions around the climate change issue.

We manually analyzed each induced structure, and assessed to which category of the codebook it could map to (see Table 2). Some structures were categorized both as a story theme (category 4) and the main theme or consequences (category 9). We also generated a category "Other" that contains all the structures that could not be categorized: grammatical structures containing only verbs or function words, geographical references (e.g. world and country), and incomplete structures.

The majority of the induced structures map to category *4*, some examples are: "*((serious|dangerous|larger|global| damaging) climate change)*", "*((the effort|the fight) against climate change)*" and "*((that (climate change*

1. ((((kan|og|må|skal|å) (overleve|bekjempe|takle))) klimaendring)
((((can|and|must|will|to) (survive|combat|tackle))) climate change)
2. ((å (tåle|unngå)) ((alvorlige|farlige|større|globale|skadelige) klimaendring))
((to (endure|avoid)) ((serious|dangerous|larger|global|damaging) climate change))
3. ((at (klimaendring er)) (naturlige|farlige|menneskeskapte))
((that (climate change is)) (natural|dangerous|man-made))
4. ((den globale) oppvarmingen)
((the global) warming)
5. ((enighet|usikkerhet|spådommer|tvil|overtydd|konklusjon|klimaforskarane|klimapanel) om)
((consensus|uncertainty|predictions|doubt|convince|conclusion|climatologists|climate panel) if)
6. ((innsatsen|kampen) mot klimaendring))
((effort|fight) against climate change)
7. ((virkningene|konsekvensen|konsekvenser) (av klimaendring))
((effects|consequence|consequences) (climate change))
8. (av hva forskerne (mener (((det er)|er) nødvendig) for (((å (tåle|unngå))|forhindre) farlige) og uopprettelige))
(of what scientists (mean ((((it is)|is) necessary) (((to (endure|avoid))|prevent) dangerous) and irreversible))
9. (dagens løfter om utslippsreduksjoner er altfor lite ambisiøse ((for|til) (å (forhindre|hindre|stanse))))
(current promises to reduce emissions is too little ambitious ((for|to) ((prevent|hinder|stop))))
10. (de (negative|første|naturvitenskapelige) effektene (av klimaendring))
(the (negative|first|scientific) effects (of climate change))
11. (fns rammekonvensjon)
(united nations framework convention)
12. (hans familie få medhold i sin søknad om flyktningstatus (på (virkningene|grunn|grunnlag|toppen) (av klimaendring)) som truer hjemlandet)
(his family get successful in their application for refugee status (on the (effects|reason|basis|top) (of climate change)) that threatens their homeland)
13. (mer ekstremvær)
(more extreme weather)
14. (sannsynligvis har (menneskelig aktivitet) ført til akutte klimaendring)
(probably have (human activity) led to acute climate change)
15. (i stand (((til å)|(med å)) ((tilpasse seg)|møte))) *(able (((to)|(to)) ((adapt)|meet)))*

Table 1: A selection of automatically induced structures around the key term "*klimaendring**" (climate change).

is)) (natural|dangerous|man-made))" . The induced structures can be divided into various keyphrases that can uncover the different aspects of a story theme.

Many structures were both categorized as category *4* and category *9*. This latter included many structures stating the various consequences of climate change, the economic status of rich and poor countries, as well as political consequences. Examples of this are: " *((serious|dangerous|larger|global|damaging) climate change)*" and "*(the (negative|first|scientific) effects (of climate change))*".

4. Story themes	*7. All the quoted voices*	*9. Main theme / Consequences*	*10. Extreme weather and natural disasters*	*Other*
159	17	53	12	155

Table 2: Amount of structures that can be mapped to the codebook's categories.

Category *10* include structures reflecting keyphrases mentioning all types of extreme weather and natural disasters as storm surges and rising sea levels, e.g.: "*(more extreme weather)*", "*(developing countries (such as ((the magnitude|denial|the consequences|following|affected|experience|worsen) of)) natural disasters linked)*" and "*(contrast to all the talk about storm surges and rising sea levels)*".

Category *7* encompasses voices present in the induced structures, but it is not possible to determine only by analyzing the structures, if all the voices have been induced. The structures provide us with an overview of the voices, but does not tell us who says what to whom. It would be beneficial to add to this analysis a simple named entity recognition process, that will help identifying all the voices, either quoted or cited, which will also enable to evaluate the strength of the automatic induction of structures with regards to important actors expressing their voices in a large corpus. Some voices that were automatically induced from the corpus are: "*(*IPCC)", "*(united nations framework convention)*", "*(the world's climate scientists)*", "*((meteorological institute))*", "*(al gore)*" and "*(president barack obama)*".

A further analysis of the structures identified in the category "Other" is necessary to uncover their correct mapping to the previous categories. We believe that an analysis of some of their concordance lines will enable the understanding of the content. Some structures categorized as "Other" are: "*((when|as) applicable)*", "*((the purpose|the target|the project) is to)*", "*((it's) (worthwhile|important) to)*" and "*(if the (world|one|we) fail to)*".

5 Concluding remarks and future work

The results presented in this paper suggest that a list of automatically induced structures, representing keyphrases, reflect some of the distinctive contents of a large corpus. Furthermore, some structures uncovered linguistic patterning that would be of interest for further investigations. In this work we have the expertise of both automatically analyzing data and manually evaluating it, which is a fruitful collaboration when merging two fields.

Several scholars have dealt with issues concerning the use of traditional content analysis on large amounts of online

news data. However, the difficulties of conducting a traditional manual content analysis on online news with traditional methods have not yet been properly discussed (with a few exceptions see e.g. [Karlsson and Sjøvaag, 2015]). We believe that our method is a step forward to develop and expand traditional content analysis within media studies, such that it can be applied on large data sets, without the need to perform time consuming manual analysis.

We were able to compress a corpus of 11.000 news articles, comprising 19.186 sentences mentioning "*klimaendring**" (climate change) into a small set of 359 structures. We only focused on a small portion of the corpus (around the key term climate change), but since the induction process is unsupervised, we believe that this approach can be applicable to various types of corpora.

The different keyphrases present in the breakdown of the structures' V-groups showed that the structures can identify some of the most important information present in a large corpus, and clarify the content of a corpus by providing a small overview, kind of summary, characterizing the content of the corpus.

With regards to understanding what a news corpus is about, and what is being discussed about important key terms, our induced structures showed their usefulness in capturing terms and phrases, and providing small overviews that enable finer-grained analysis, classification, indexing and text retrieval from a large corpus.

An additional apparent advantage of the induced structures is their ability to group together alternative keyphrases that refer, in different ways, to the same concept. We can see this as a potential to uncover how an issue is framed differently from the various angles of a discussed issue. But we also see the need to improve the method to reduce the amount of structures classified as "Other".

We have proposed and evaluated a novel idea of using automatically induced structures as keyphrases for analyzing large news corpora. We foresee media scholars, and journalists, exploring a list of induced structures as a first step to get an overview of the content of large corpora, and to identify interesting phenomena for further detailed analysis. We believe that our method can shed light on aspects of the content that cannot be easily identified manually. This work indicates that our method facilitates and improves the quality of the analysis, the quantity of the data analyzed, and offers a deeper understanding of a large corpus.

In future work, we aim to further develop the method in order to understand and reduce the amount of structures categorized as "Other". We believe that a thorough analysis of the context of these structures will enable us to categorize them into the appropriate codebook's categories; such that an analysis of their concordance lines, and their collocates, should give a sufficient description of their context.

We also aim to develop an approach that will enable a better identification of the voices present in a news article. As mentioned earlier, it would be beneficial to add a named entity recognition process, that will help identifying all the voices that have been referenced. It would also be interesting to be able to uncover who says what, and to whom. This needs a more sophisticated computational approach that can identify and extract the actors and their "voices". However, both for media scholars and journalists it is far more important to define who are the sources in the newspaper coverage, which are not definable using our method.

In addition, it would be interesting to compare our method to well established methods for extracting keyphrases, as well as LDA (Latent Dirichlet Allocation [Blei *et al.*, 2003]) which have been extensively used for content analysis. The comparison should be made on common grounds, and using human judges to evaluate the outputs of the different methods in use.

References

[Blei *et al.*, 2003] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[Boykoff and Boykoff, 2007] Maxwell T. Boykoff and Jules M. Boykoff. Climate change and journalistic norms: A case-study of {US} mass-media coverage. *Geoforum*, 38(6):1190 – 1204, 2007. Theme Issue: Geographies of Generosity.

[Boykoff *et al.*, 2015] M Boykoff, M Daly, L Gifford, G Luedecke, L McAllister, A Nacu-Schmidt, and K Andrews. World newspaper coverage of climate change or global warming, 2004–2015. *Center for Science and Technology Policy Research, Cooperative Institute for Research in Environmental Sciences, University of Colorado, online, accessed*, 10, 2015.

[Carvalho and Burgess, 2005] Anabela Carvalho and Jacquelin Burgess. Cultural circuits of climate change in uk broadsheet newspapers, 1985–2003. *Risk analysis*, 25(6):1457–1469, 2005.

[D'Ulizia *et al.*, 2011] Arianna D'Ulizia, Fernando Ferri, and Patrizia Grifoni. A survey of grammatical inference methods for natural language learning. *Artificial Intelligence Review*, 36(1):1–27, 2011.

[Eide and Kunelius, 2012] Elisabeth Eide and Risto Kunelius. *Media meets climate : the global challenge for journalism*. NORDICOM, Göteborg, 2012.

[Grimmer and Stewart, 2013] Justin Grimmer and Brandon M Stewart. Text as data: The promise and pitfalls of automatic content analysis methods for political texts. *Political Analysis*, page mps028, 2013.

[Harris, 1954] Zellig S Harris. Distributional structure. *Word*, 1954.

[Hasan and Ng, 2014] Kazi Saidul Hasan and Vincent Ng. Automatic keyphrase extraction: A survey of the state of the art. In *ACL (1)*, pages 1262–1273, 2014.

[Karlsson and Sjøvaag, 2015] Michael Karlsson and Helle Sjøvaag. Content Analysis and Online News: Epistemologies of analysing the ephemeral Web. *Digital Journalism*, 4(1):177–192, 2015.

[Krippendorff, 2004] Klaus Krippendorff. *Content analysis : an introduction to its methodology*. Sage, Thousand Oaks, Calif, 2nd ed. edition, 2004.

[Lamb, 1961] Sydney Lamb. On the Mechanization of Syntactic Analysis. *Int. Conf. Machine Translation of Languages and Applied Language Analysis*, 1961.

[Liu *et al.*, 2011] Xinsheng Liu, Eric Lindquist, and Arnold Vedlitz. Explaining media and congressional attention to global climate change, 1969-2005: an empirical test of agenda-setting theory. *Political Research Quarterly*, 64(2):405–419, 2011.

[Neuendorf, 2002] Kimberley A. Neuendorf. *The content analysis guidebook : Kimberly A. Neuendorf*. Sage, Thousand Oaks, Calif, 2002. Bibliografi: s. 247-282.

[Salway and Touileb, 2014] Andrew Salway and Samia Touileb. Applying grammar induction to text mining. In *ACL (2)*, pages 712–717, 2014.

[Schmidt *et al.*, 2013] Andreas Schmidt, Ana Ivanova, and Mike S Schäfer. Media attention for climate change around the world: A comparative analysis of newspaper coverage in 27 countries. *Global Environmental Change*, 23(5):1233–1248, 2013.

[Shaw, 2013] Christopher Shaw. Choosing a dangerous limit for climate change: Public representations of the decision making process. *Global Environmental Change*, 23(2):563 – 571, 2013.

[Solan *et al.*, 2005] Zach Solan, David Horn, Eytan Ruppin, and Shimon Edelman. Unsupervised learning of natural languages. *Proceedings of the National Academy of Sciences of the United States of America*, 102(33):11629–11634, 2005.

[Touileb and Salway, 2014] Samia Touileb and Andrew Salway. Constructions: a new unit of analysis for corpus-based discourse analysis. In *Proceedings of the 28th Pacific Asia Conference on Language, Information and Computation*, 2014.

[Turney, 2000] Peter D Turney. Learning algorithms for keyphrase extraction. *Information Retrieval*, 2(4):303–336, 2000.

A multi-lingually applicable journalist toolset for the big-data era

G. Kiomourtzis and **G. Giannakopoulos** and **V. Karkaletsis**
NCSR "Demokritos", Athens, Greece
{ggianna|gkiom|vangelis}@iit.demokritos.gr

A. Kosmopoulos
SciFY PNPC, Athens, Greece
akosmo@scify.org

Abstract

This paper overviews the NewSum Toolkit toolset, providing a full set of Natural Language Processing tools aimed to support journalists and publishers during the news writing phase. The toolset contains a set of services, currently integrated in commercial products, that can overview and summarize thousands of sources (social media and news feeds) in different languages, support automatic article classification based on existing or new taxonomies, and facilitate slug-line generation. The overall system builds on well-established, open technologies and tools to provide a backbone that can empower any publishing platform.

1 Introduction and Related work

Several advances in sub-domains of Natural Language Processing (NLP) have, for several years, been overlooked by the media industry, because oftentimes the efficiency of the tools was insufficient to be applied in a real-world, open domain setting. Recent years have, however, redefined the relation between NLP and the world of media. The redefinition is related to both the changes in the mass media landscape [Curran, 2010] — leading to data journalism, crowdsourced [Brabham, 2012] and crowdfunded journalism [Aitamurto, 2011] — but also the updated focus of scientific efforts and results towards real world and industrial problems: multi-document news summarization [Dang and Owczarzak, 2008; Giannakopoulos *et al.*, 2011], argument summarization [Swanson *et al.*, 2015], forum and socal media summarization [Kabadjov *et al.*, 2015] to scientific summarization [Qazvinian *et al.*, 2013].

Research has led a number of efforts to empower news summarization, with the Document Understanding and Text Analysis Conferences (e.g. [Dang, 2006; Dang and Owczarzak, 2008]) and with NTCIR[1]. Throughout the years, many different approaches on summarization have been proposed, ranging from seminal works on simple, frequency or keyword based schemas [Luhn, 1958] to latent semantic spaces and network-based methods [Mihalcea, 2005]. However little has been done to empower the journalist. Even though there exist several online tools related to summarization, most focus on the reader-consumer or to automatic summarization per se. Such tools include search result summaries (e.g. JistWeb and Ultimate Search Assistant), and single document summarizers (TLDR Reader, ReadBorg).

In this work we overview NewSum Toolkit, which comes to address this exact need: bringing reusable, state-of-the-art Natural Language Processing to the journalist, empowering news writing and publishing. We then conclude, providing a glimpse of future extensions to the toolkit.

[1] See http://research.nii.ac.jp/ntcir/ for more information.

2 NewSum Toolkit: An overview

The NewSum Toolkit toolset can be broken down into a number of services that cover a variety of journalistic needs, as follows.

Data gathering This service provides a number of web services that allow management and monitoring of media sources. These sources can be news feeds (e.g. RSS/Atom feeds), blogs and websites or social media sources (e.g. Twitter, Facebook). This service allows journalists to follow all their sources of interest under a unified view.

Summarization The summarization component of the system, implements two critical functions. First, it identifies events, as these are reported across sources. To this end, a number of NLP methods are combined to first measure the similarity and then group content items (e.g. posts, articles) into events. The second function this component achieves is that of summarization. Essentially, employing language-agnostic summarization methods based on open n-gram graph technologies [Giannakopoulos *et al.*, 2014], the system undertakes the task to provide a snippet-based (extractive) summary of each event. This summary aims to provide representative information, while removing the expected (and often extreme) redundancy from the varying sources. The system ascertains that the summary contains all the links back to the original sources, allowing verification and minimizing error.

Trend detection In the trend detection component, events are followed throughout their life-span to detect importance over time. Using varying windows of time, an importance index is calculated, which represents overall

and instantaneous trend. The output of this component allow for the ranking of incoming events and can help the journalists prioritize their focus.

Automatic Classification Within the news generation process, there exists a number of near-trivial tasks that can take significant time, such as the annotation of news with specific classification codes (e.g. based on the International Press Telecommunications Council or IPTC media topics classification[2]). To help journalists, the automatic classification component can suggest categories and annotations, minimizing human effort. This component employs machine learning techniques to improve over time, based on the actual annotations and corrections of the users.

Slugline generation Sluglines are another type of repetitive and time-consuming task that NewSum Toolkit comes to support: a slugline contains a few keywords and phrases, aiming to outline the setting of an event. Based on named entity recognition and keyword extraction methods, the system undertakes the task of suggesting a slugline, which can then be finalized by the journalist.

Overall, the above set of tools have been created to facilitate the whole life-cycle of news generation, from the identification of important (or potentially important) events, to the actual writing of the article. The system is built to be applicable over a large range of languages, with minimal fine-tuning, since it relies on mostly language-agnostic approaches. Furthermore, each component has been implemented with a parallel execution approach and easy cloud-based integration, making the system "big-data"-enabled and allowing scaling to tens of thousands of sources with minimum effort.

In the following paragraphs we elaborate on the NewSum Toolkit components, providing some insights on the related technical approach and provided outputs.

2.1 Data gathering

The data gathering service supports RSS/Atom feeds, blogs, web pages scraping (through customization), and different kinds of social media, i.e. Twitter, Facebook, etc. The service is designed in an easily extensible way, to support other kinds of sources with ease. The news/blogs/web sites module operates regularly and updates the database with new entries, avoiding duplication, while the social media modules are targeted on a supplied — by the journalist — group of accounts, which are monitored in short time intervals.

The module is known to operate well on an 8-CPU core server with 16GB of RAM, in a setup of more than 1000 sources (including 200 monitored social media accounts) with a 15 minute refresh interval. The news module uses a distributed NoSQL solution (mongoDB, cf. [Abramova and Bernardino, 2013]) as a persistence mechanism, while the social media data are temporarily stored in a relational database — due to a more relational structure of the content. Further processing unifies the metadata in mongoDB to improve scalability.

[2]See http://cv.iptc.org/newscodes/ for more information regarding IPTC codes.

The RSS/blog/web sites module is built using open source Java libraries, covering RSS feed parsing and HTML page (DOM) parsing. For the social media data gathering, NewSum Toolkit utilizes the corresponding most popular open source Java libraries for each respective medium (Twitter, Facebook, etc.), which build upon the corresponding REST APIs (e.g. Twitter API, Facebook Graph API).

Once again, the data gathering is a highly parallelizable process, which allows for scalability in both storage and execution.

2.2 Summarization

Summarization in NewSum Toolkit is a three stage process. In the first stage, documents are grouped into clusters that we expect will refer to one topic or event. In the second stage, the systems processes each cluster, determining "sub-topics" — i.e. aspect of the topic/event discussed in the cluster documents. Then, each subtopic is represented by a set of sentences that maximally cover the subtopic, with minimal redundancy.

The clustering process can be broken down into the following individual steps: meta-data-based document grouping; similarity calculation within groups; determination of clusters. The steps are elaborated below.

Initially, documents are mapped to groups according to their meta-data, i.e. items from the same news category, etc. Essentially, this is a blocking step to reduce the complexity of pairwise comparisons between documents and allow efficient, large scale analysis. This approach also covers a user need which connects specific sources to be appropriate for specific news categories (e.g. foreign policy news vs. music industry news). The meta-data assigned to each source are provided during the setup phase of the NewSum Toolkit. Other meta-data may originate from the item itself (e.g. category specification in the NewsML format of a news item).

For each item pair, specific similarity measurements are extracted, which are then combined with heuristic rules to determine pairs of related documents. The similarity measurements use pre-processing and text analysis to extract a number of features, related to named entities and morphological characteristics of the text.

Given the results of this pairwise matching, the system determines the transitive closure of the matching relation to define clusters. In other words, a graph is generated between documents, where edges depict a "refer-to-the-same-event" relation between documents. Whenever there exists a path between two documents in this graph, the documents belong to the same cluster. Thus, the result of the clustering is a hard (i.e. non-overlapping) set of clusters, where each cluster of documents is considered to represent documents referring to the same event (cf. Figure 1).

For each topic/event cluster, we then identify what we term "sub-topics". A sub-topic is a cluster of sentences that is expected to reflect a specific view of a topic (e.g. the financial vs. the political aspect). To determine these sub-topics, we first represent sentences as character n-gram graphs [Giannakopoulos *et al.*, 2008]. We then compare all pairs of sentences, based on their character n-gram representation. We use character 3-grams and a neighborhood distance of 3 to

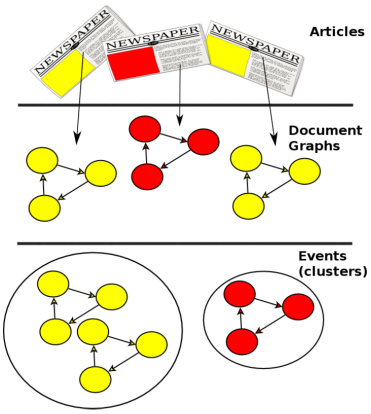


Figure 1: Clustering for event detection

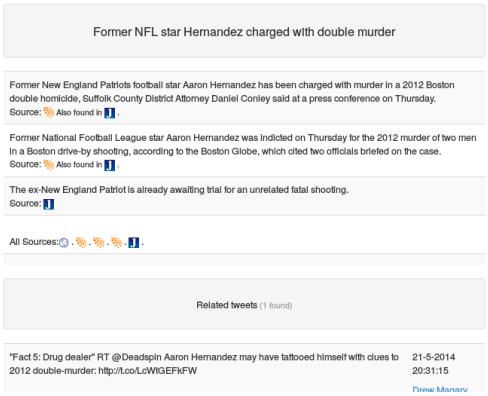


Figure 2: A cross-media news summary

represent the sentences. These values have been shown to perform well in a variety of settings. The output of this step is a similarity matrix between sentences, based on the Normalized Value Similarity (NVS) between the n-gram graphs of the sentence pairs [Giannakopoulos *et al.*, 2008].

We then apply Markov Clustering (MCL) [Dongen, 2000] on the similarity matrix. The result of this process is a set of hard clusters, identifying sub-topics. The summarization process concludes by selecting sentences that are most representative for the sub-topic, while limiting redundancy. To achieve this double goal, we first create representative "centroid" graphs per topic. We then form n-gram graphs for every candidate sentence. Then we sort candidate sentences based on the similarity of their graph to the representative graph of the sub-topic. Then, we run through the sorted candidate list, removing sentences that appear too similar (i.e. surpass a similarity threshold between each other).

The above method has been used in the MultiLing, multilingual multi-document, summarization challenge [Giannakopoulos *et al.*, 2015] with promising results (ranking in the top 50% of systems in the overall ranking across all languages) without fine-tuning. In the industrial system we apply fine-tuning to increase the performance on target languages.

We note that the system also holds the potential to link clusters between the news (RSS feeds) and the social media (e.g. Twitter). To this end, we employ n-gram graph similarity between the news and the social media clusters; if the similarity exceeds a heuristically-defined threshold, we connect the clusters as referring to the same topic. We illustrate such an example in Figure 2.

2.3 Trend detection

In order for the suite to detect trending events, it groups the news clusters (i.e. topics) into an abstraction defined as "story". Essentially, a story depicts the way topics are evolving in a time period. The significance of a specific event is closely related to the news sources that write about this event, i.e. the topic size. The trend detection algorithm operates in two ways: first it extracts the number of sources that determine a story in the time window that the story exists, and secondly it looks for instantaneous trend by only checking the topic size of the two most recent topics of the story. These two approaches are meant to indicate two different types of trends: instantaneous trend, which implies topic "hotness", but also overall trend, which implies robustness in the significance of topic. Essentially, the trend if the delta of the number of sources either across two individual time-points (instantaneous trend) or over a span of time (e.g. two days).

The above approaches allow the journalist to shift the focus from *potentially* important topics with reduced coverage — i.e. topics with significant instantaneous trend — to topics that persistently remain important over time, as appropriate.

2.4 Automatic classification

The goal of the automatic classification component is to suggest IPTC codes (i.e. topic-related codes from a hierarchy of formalized classes) for each news article. Our system uses one binary classifier for each class (IPTC code). During the prediction step a probability is computed for each class in order to suggest the IPTC codes with the highest probability to the journalist.

Annotated NewsML[3] articles are used for training the classifiers for each class. We first extract the text from the *NewsLineText* and *DataContent* tags of the NewsML file. We then transform the extracted text to feature vectors using a bag of words approach. Stemming procedures and TF-IDF transformation are also used. Adapted strategies regarding these prepossessing techniques take advantage of language-specific resources to support several languages (e.g.English and Greek in our current installations), but we do not elaborate on these fine-tuning processes here for lack of space.

An L2 Logistic Regression [Fan *et al.*, 2008] classifier is used to model each class. This method is quite accurate for binary classification, while also providing a probability for its prediction, which can be exploited for multi-label classification (i.e. by keeping highly confident predictions). Second it is quite scalable and since training and prediction for each

[3]See https://iptc.org/standards/ for more information regarding NewsML formats.

class is independent to the others, it can be run in parallel to improve scalability.

This process supports journalists in the annotation of their stories and minimizes the related effort.

2.5 Slugline generation

Slugline generation relies mostly on Named Entity Recognition (NER), to suggest a slugline to the journalist in the time of story writing and editing. We combine a set of entity lists (gazetteers) from a variety of sources (name/surname lists, organizations, etc.) with NER models (based on the OpenNLP toolkit [Baldridge, 2005]) to identify entities in the journalist article. Thus, the proposed slugline contains the identified set of entities. Ongoing work also builds upon keyword extraction techniques to supplement the extracted entities in the slugline and improve on our current approach.

2.6 User studies

We note that in the past [Giannakopoulos *et al.*, 2014] we had conducted user studies to identify the usefulness of the summarization, as well as the quality of the summaries.

To answer the question of whether our system (early user interface of the toolkit in 2013) facilitates news reading, we performed a small scale user experience experiment, limited to 18 Greek and 7 English beta testers [Giannakopoulos *et al.*, 2014]. These testers, who were recruited via an open call, were provided a questionnaire that measured different aspects of user experience. The question we will discuss here was expressed as "The use of NewSum allowed me to get informed of the latest news more globally and thoroughly than before". The answer allowed a 5-scale response from "I totally disagree" to "I totally agree". 11 out of 18 (61%) Greek users and 4 out of 7 (58%) English users have an answer of 4 or 5 to this question. Only 1 user per language thought that the system did not really help (2, in the 5-scaled response). The mean grade for Greek was 4 with a standard error of 0.24; for English the mean was 3.86 with a standard error of 0.46. Thus, these preliminary results indicate that users tend to believe that using the summaries can improve their news reading, fulfilling its purpose.

Two more user studies [Giannakopoulos *et al.*, 2014] on closed and open beta versions of the summarization system. The latter evaluation, which featured an improved version of the summarization system, returned 720 ratings mapped to individual users. 267 ratings were for English summaries and 453 for Greek, showing promising performance but also holding more findings. The overall rating distribution over all users and languages are shown in Figure 3.

The first finding of this study was that the language and the user are highly statistically significant factors for the results, and this was also shown by the average performances: for Greek the average was 4.14 (with a standard deviation of 1.07), while for English the average was 3.73 (with a standard deviation of 1.34). This showed that fine-tuning may make sense on individual languages and later versions of the system have this ability. In both languages the average performance was good, with more than 90% of the summaries having an acceptable (or better) grade for Greek and more than 80% for English. For a detailed description of the setting and

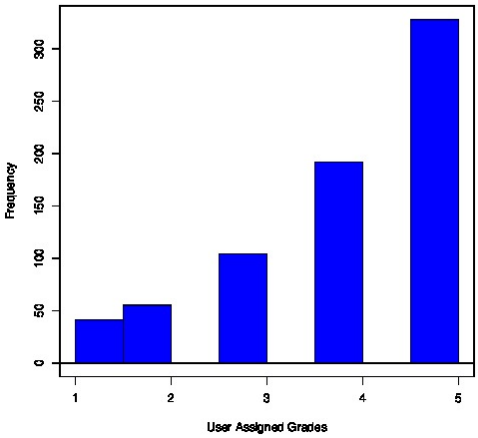


Figure 3: User study evaluation results over all languages and users

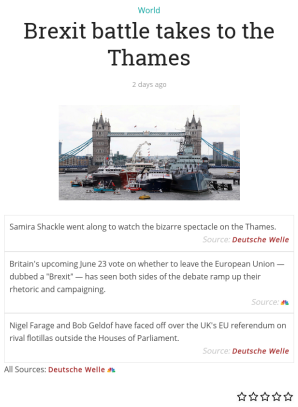


Figure 4: A summary snapshot of the web-based demonstrator of the NewSum Toolkit

findings, please consult the original work [Giannakopoulos *et al.*, 2014].

In recent interfaces of the system, such as the one displayed in Figure 4, ratings are inherently supported to further improve the performance of the system over time (cf. Figure 4). A demonstrator can be found in the following URL: `http://newsumontheweb.org/`.

2.7 Challenges faced and lessons learnt

The combination of all the above technologies under a single product holds several challenges. We briefly describe the main challenges per module.

In data gathering, there exists a variety of embedded information even in RSS feeds. This information can vary from videos to advertisements. Oftentimes, the news feed itself is a flawed implementation of protocols, which provides partially malformed data. Thus, the integration of sources usu-

ally needs human effort to both clean-up and maintain over time.

In summarization, there exist domains where the clustering similarity threshold value varies significantly. In some cases, e.g. highly focused domains such as the car industry, the identifying elements are not topic-related terms, but rather named entities (car brandsor models). Thus, one needs to support several types of clustering algorithms, based on different features. We have also understood that cluster precision (being precise in what you put in the summary) is more important than recall (adding all related conent to the summary). The perceived value of the summary is significantly undermined if one sees irrelevant text (even a single piece) in the summary.

In classification, the training data offered per category heavily affect the performance of a classification — as expected in most classification settings. Thus, the user needs to be informed on best practices for the training of new categories. Oftentimes, returning a confidence per classification decision that the system took can further improve the usefulness of the system.

Finally, in slugline generation, fine-tuning is needed to achieve a balance between precision and recall. As an example, exhaustive name lists often include names that are also location names (e.g. Thessaloniki as a location and as a person name). Pragmatic knowledge biases our human view of the entity (Thessaloniki is almost always used as a location) and such knowledge needs to be integrated into the system to avoid misclassifications.

Overall, the automated algorithms form a robust basis, which however needs adjustments to provide production-level performance per domain of application.

3 Conclusion and future work

In this paper we overview the NewSum Toolkit toolset, aimed at empowering journalists through Natural Language Processing and Machine Learning tools. The toolset supports a number of stages in the story writing, from gathering information, to detecting important topics and writing the final text. NewSum Toolkit is essentially an infrastructure that can be integrated with any platform and tool through a friendly API, taking advantage of the cloud infrastructure and scalable, parallel algorithms to cope with the big data environment.

In the future plans we include a time-line view of news, supporting the journalist in the documentation of an event, by providing easy access to past knowledge. We also foresee a real-time editor module, empowered with automatic suggestions related to writing style, and with enrichment features that will use linked open data standards to annotate articles, to improve their publication efficiency.

Acknowledgments

This paper is supported by the project "Your Data Stories – YDS", which has received funding from the European Unions Horizon 2020 research and innovation programme under grant agreement No 645886.

References

[Abramova and Bernardino, 2013] Veronika Abramova and Jorge Bernardino. Nosql databases: Mongodb vs cassandra. In *Proceedings of the International C* Conference on Computer Science and Software Engineering*, pages 14–22. ACM, 2013.

[Aitamurto, 2011] Tanja Aitamurto. The impact of crowdfunding on journalism: Case study of spot. us, a platform for community-funded reporting. *Journalism practice*, 5(4):429–445, 2011.

[Baldridge, 2005] Jason Baldridge. The opennlp project. *URL: http://opennlp. apache. org/index. html,(accessed 2 February 2012)*, 2005.

[Brabham, 2012] Daren C Brabham. The myth of amateur crowds: A critical discourse analysis of crowdsourcing coverage. *Information, Communication & Society*, 15(3):394–410, 2012.

[Curran, 2010] James Curran. The future of journalism. *Journalism studies*, 11(4):464–476, 2010.

[Dang and Owczarzak, 2008] H. T Dang and K. Owczarzak. Overview of the tac 2008 update summarization task. In *TAC 2008 Workshop - Notebook papers and results*, page 1023, Nov 2008.

[Dang, 2006] H. T Dang. Overview of duc 2006. In *Proceedings of HLT-NAACL 2006*, 2006.

[Dongen, 2000] Stijn Dongen. Performance criteria for graph clustering and markov cluster experiments. 2000.

[Fan *et al.*, 2008] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[Giannakopoulos *et al.*, 2008] George Giannakopoulos, Vangelis Karkaletsis, George Vouros, and Panagiotis Stamatopoulos. Summarization system evaluation revisited: N-gram graphs. *ACM Transactions on Speech and Language Processing (TSLP)*, 5(3):5, 2008.

[Giannakopoulos *et al.*, 2011] George Giannakopoulos, Mahmoud El-Haj, Benoit Favre, Marina Litvak, Josef Steinberger, and Vasudeva Varma. Tac2011 multiling pilot overview. In *TAC 2011 Workshop*, 2011.

[Giannakopoulos *et al.*, 2014] George Giannakopoulos, George Kiomourtzis, and Vangelis Karkaletsis. *NewSum: "N-Gram Graph"-Based Summarization in the Real World*. IGI, 2014.

[Giannakopoulos *et al.*, 2015] George Giannakopoulos, Jeff Kubina, Ft Meade, John M Conroy, MD Bowie, Josef Steinberger, Benoit Favre, Mijail Kabadjov, Udo Kruschwitz, and Massimo Poesio. Multiling 2015: Multilingual summarization of single and multi-documents, online fora, and call-center conversations. In *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 270, 2015.

[Kabadjov *et al.*, 2015] Mijail Kabadjov, Josef Steinberger, Emma Barker, Udo Kruschwitz, and Massimo Poesio. Onforums: The shared task on online forum summarisation at multiling15. *Analysis*, 16:23, 2015.

[Luhn, 1958] H. P Luhn. Automatic creation of literature abstracts, the. *IBM Journal of Research and Development*, 2(2):159165, 1958.

[Mihalcea, 2005] R. Mihalcea. Multi-document summarization with iterative graph-based algorithms. In *Proceedings of the First International Conference on Intelligent Analysis Methods and Tools (IA 2005)*. McLean, 2005.

[Qazvinian *et al.*, 2013] Vahed Qazvinian, Dragomir R. Radev, Saif M. Mohammad, Bonnie Dorr, David Zajic, Michael Whidby, and Taesun Moon. Generating extractive summaries of scientific paradigms. *Journal of Artificial Intelligence Research*, 46:165201, 2013.

[Swanson *et al.*, 2015] Reid Swanson, Brian Ecker, and Marilyn Walker. Argument mining: Extracting arguments from online dialogue. In *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 217, 2015.

A Computational Approach to the Study of Portuguese Newspapers Published in Macau

Marcos Zampieri[1,2], Shervin Malmasi[3], Octavia-Maria Şulea[1,2,4], Liviu P. Dinu[4]
Saarland University, Germany[1]
German Research Center for Artificial Intelligence (DFKI)[2]
Harvard Medical School, Boston, MA, USA[3]
University of Bucharest, Romania[4]

Abstract

This paper investigates the application of text classification methods to investigate diatopic variation in Portuguese journalistic texts. We compare the language used in Portuguese newspapers written in Brazil, Macau, and Portugal under the assumption that the more similar language varieties are, the more difficult it is for algorithms to discriminate between them. We present two sets of experiments: in the first one we use original texts and in the second one we use texts with blinded named entities to remove country-specific expressions. Our results indicate that the language of Portuguese newspapers published in Macau is substantially more similar to the language used in European newspapers than that used in Brazilian newspapers.

1 Introduction

Portuguese is a pluricentric language (co-)official language in nin countries and in Macau, a special administrative region of China with a population of around 650,000 people. Portuguese is spoken by a minority of the Macanese population (between 5% and 10%). It coexists with Cantonese, spoken by over 90% of the population, and Macanese, a Portuguese-based creole. In Macau, Portuguese is used for official communication in street signs, official documents, and media including a Lusophone TV channel, radios, and newspapers.

In this paper we propose the use of text classification methods to study the language used in Portuguese newspapers published in Macau in comparison to newspapers published in other Lusophone countries, namely: Brazil and Portugal. There have been a number of studies on the differences between Portuguese language varieties and on Portuguese and Portuguese-based creoles in Macau [Baxter, 1992; 1996; Amaro, 2016], however, to the best our knowledge, no study has been carried out in order to investigate the current language of journalism in Macau. A similar study [Zampieri and Gebre, 2012] has shown that Brazilian and European newspaper texts use substantially different language and that a system trained on character and word n-grams can distinguish between them with 99.8% accuracy.

Our work is related to recent studies which apply text classification methods for discriminating between texts written in different national language varieties or dialects [Lui and Cook, 2013; Maier and Gómez-Rodrıguez, 2014; Malmasi and Dras, 2015a; Malmasi *et al.*, 2015]. It has been argued that such experiments are useful to level out differences between corpora for further linguistic analysis [Zampieri *et al.*, 2013; Ciobanu and Dinu, 2016]. We agree with this claim and we analyze the most informative lexical features used in our experiments in Section 4.1.

The question we aim to answer in this paper is:

- Are there substantial differences between the language used in newspapers published in Macau and those published in other Lusophone countries?

In addition to the historical cooperation and exchange between Macau and Portugal in areas such as trade and culture, many Portuguese speakers currently living in Macau are actually Portuguese expats. The hypothesis we would like to test is whether, despite the great geographical distance between Macau and Portugal, both of the aforementioned factors influence journalists based in Macau to use a language that is similar to the European Portuguese standard.

2 Methods

2.1 Corpus

In this paper, we use the three Portuguese sub-corpora from Brazil, Macau, and Portugal (hereafter BR, MO, and PT) available in the dataset of the 2015 edition of the Discriminating between Similar Languages (DSL) shared task [Zampieri *et al.*, 2015], the DSL Corpus Collection (DSLCC) version 2.1 [Tan *et al.*, 2014].

The DSLCC is a collection of journalistic texts compiled from multiple sources, including previously released corpora, containing short text excerpts sampled from various newspapers.[1] According to the information provided by the authors of the DSLCC, Macanese texts were compiled from two newspapers: *Tribuna de Macau* and *Hoje Macau*.[2,3]

The three Portuguese sub-corpora combined contain a total of 54,000 excerpts (documents) and each document contains between 20 and 100 tokens. Table 1 presents the number of documents and types in each sub-corpus.

[1] A list of sources is available in [Tan *et al.*, 2014].

[2] http://jtm.com.mo/

[3] http://hojemacau.com.mo/

	Tokens	Types	Documents
BR	602,684	41,419	18,000
MO	547,479	32,547	18,000
PT	582,420	36,313	18,000
Total	1,732,583	-	54,000

Table 1: Number of Tokens, Types, and Documents in the DSLCC BR, MO, PT sub-corpora

2.2 Computational Approach

We approach the task using a text classification system based on a linear SVM classifier implemented in LIBLINEAR [Fan *et al.*, 2008]. SVMs proved to deliver very good performance in discriminating between language varieties, achieving first place in both the 2015 [Malmasi and Dras, 2015b] and 2014 [Goutte *et al.*, 2014] editions of the DSL shared task.[4]

We use unigram and bigram language models to capture lexical and lexico-syntactic differences between the newspapers published Brazil, Macau, and Portugal. Unigram language models have been used to discriminate between Brazilian and European Portuguese newspapers texts with results of over 99% accuracy by [Zampieri and Gebre, 2012].[5] In this study researchers pointed out that lexical variation and orthographic differences play an important role in the task.

We evaluate the performance of our method using standard NLP evaluation metrics such as precision (P), recall (R), and f-score (F) for multi-class classification and accuracy (A) for binary classification settings. All results are presented using k-fold cross-validation, with $k = 10$. We consider random baseline as the baseline performance for this task. This means 33% accuracy for classification sets containing three classes and 50% accuracy for binary classification settings.

3 Results

In our first experiment, we apply the aforementioned SVM classifier to discriminate between the three corpora. We report precision, recall, and f-score in Table 2.

Features	Class	P	R	F
Unigrams	BR	0.87	0.90	0.88
	MO	0.78	0.76	0.77
	PT	0.76	0.75	0.75
	Average Scores	0.80	0.80	0.80
Bigrams	BR	0.82	0.91	0.86
	MO	0.78	0.72	0.75
	PT	0.74	0.72	0.73
	Average Scores	0.78	0.78	0.78

Table 2: Three-way classification (BR, MO, PT)

The classifier achieves an average performance of 80% f-score using unigrams and 78% f-score using bigrams suggesting that the lexical differences are the most informative information in this task. We observed that performance varies substantially among the classes. Using unigrams performance is higher for the Brazilian class (88% f-score) than for Macau (77% f-score) and Portugal (75% f-score). We investigate the performance variation by analyzing the confusion matrix of the word unigram results in Figure 1.

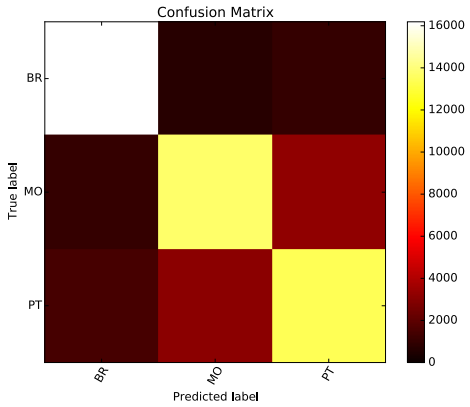


Figure 1: Confusion matrix for three-way classification

The confusion matrix shows that there is substantial confusion between MO and PT texts whereas BR texts are always the easiest to identify. To investigate this further we conduct binary classification experiments in which the algorithm is trained to choose between texts from only two countries at a time. We present accuracy results in Table 3.

Features	Newspapers	Accuracy
Unigrams	BR vs. PT	0.91
	BR vs. MO	0.93
	MO vs. PT	0.79
Bigrams	BR vs. PT	0.89
	BR vs. MO	0.91
	MO vs. PT	0.77

Table 3: Binary classification results

The algorithm achieves very good performance, 91% accuracy, when discriminating between BR and PT texts using unigrams which corroborates the findings by [Zampieri and Gebre, 2012]. The method achieves even higher performance, 93% accuracy, discriminating between BR and MO texts using unigrams. We investigate the reasons for this high performance in Section 4.1. The performance of the classifier discriminating between MO and PT texts is substantially lower than the other two. These outcomes confirm our hypothesis that currently Macanese newspaper texts are similar to the European Portuguese standard.

It is well known that named entities (NE) such as people, places, and organization play an important role in this task. To investigate the influence of NEs in classification we propose a second round of experiments presented next section.

3.1 The Influence of Named Entities

It is safe to assume that texts published in Portugal are more likely to refer to *Lisbon* and to the *European Union* than

[4] See [Goutte *et al.*, 2016] for a comprehensive evaluation.

[5] It should be noted that in [Zampieri and Gebre, 2012] the authors use full texts containing up to 500 tokens.

Macanese texts and that texts from Brazil are very likely to include names of famous Brazilian people and places. We observed this phenomenon in the analysis of the most informative features obtained in the experiments described in Section 3. To diminish the influence of country-specific expressions in classification we replicate an experiment proposed in the second edition of the DSL shared task. The experiment consists of substituting most named entities in text by placeholders *#NE#*. The DSL approach to named entity removal addresses only capitalized proper nouns and all words which are not capitalized are left in text.[6] Below is an example of how texts are represented before and after this substitution:

(1) Compara este sistema às indulgências vendidas pelo Clero na Idade Média, quando os fiéis compravam a redenção das suas almas dando dinheiro aos padres.

(2) Compara este sistema às indulgências vendidas pelo #NE# na #NE# #NE# quando os fiéis compravam a redenção das suas almas dando dinheiro aos padres.

We use texts produced after NE substitution firstly in a three way classification setting involving BR, MO, and PT texts. We used word unigrams as features because these were the best performing features presented in the last section. In Table 4 we include a column *Diff.* which contains the difference between the f-scores obtained using original texts and texts with NE substitution in percentage points.

	P	R	F	Diff.
BR	0.83	0.86	0.84	-4 pp
MO	0.70	0.72	0.71	-6 pp
PT	0.70	0.65	0.67	-8 pp
Average Scores	0.74	0.74	0.74	-6 pp

Table 4: Blind NE - Three-way classification (BR, MO, PT)

In this setting we observed that BR texts were again the easiest to identify. However, the performance of the algorithm identifying PT texts dropped 8 percentage points. We repeat the binary experiments without NEs and present results in terms of accuracy in Table 5.

Newspapers	Accuracy	Diff.
BR vs. PT	0.88	-3 pp
BR vs. MO	0.90	-3 pp
MO vs. PT	0.74	-5 pp

Table 5: Blind NE - Binary classification results

The results obtained by the classifier when discriminating between MO and PT texts were worse than when using the settings featuring the BR class (Table 5). Moreover, we observed a performance drop of 3 percentage points for the two settings including BR texts whereas the result obtained by the algorithm discriminating between MO and PT texts without NE were 5 percentage points lower. This once again suggests that BR texts are substantially different from both MO and PT texts. Finally, we observed a performance drop from 3 to 8 percentage points in all settings which confirms that named entities play an important role in this task. However, as mentioned earlier in this section, not all named entities were removed from texts, and in the most informative features we find a number of expressions which are country-specific. We discuss this in more detail in the next section.

[6]We used the script provided by the DSL shared task organizers: https://github.com/alvations/bayesmax/tree/master/bayesmax

4 Discussion

4.1 Feature Analysis

Our results show that BR newspapers can be identified with over 90% accuracy. Our analysis of the most informative features indicate that this is mostly due to orthographic conventions, for example mute consonants used in Portugal *(director)* and not used in Brazil *(diretor)*, and because of words that are more frequently used in Brazil than in other Portuguese speaking countries, for example *você* (EN: *you*). The top ten most informative features are presented in Table 6.

The method discriminates between texts from PT and MO using original texts and texts without most NEs with 79% and 74% accuracy respectively. This represents above chance (50%) performance, but it is still substantially worse than the performance obtained by the classifier when discriminating between texts from BR and MO or BR and PT. According to our feature analysis, texts from PT and MO were identified mostly relying on country-specific words. Using original texts, half of the top ten most informative features in PT texts are names of Portuguese regions or cities such as *Leiria, Aveiro, Braga, Algarve*, and *Minho* are among the top-10 most informative lexical features in MO texts after NE removal we find *patacas* (the Macanese currency), *chinesa*, and *macaense*, all of them country-specific.

Answering the question posed in the introduction, our classification results and the analysis of the most informative features suggest that there are substantial differences between the language used in BR and MO newspapers in terms of orthography and lexicon, but not between MO and PT texts. Although the SVM classifier was able to discriminate between MO and PT texts with above chance performance, the algorithm did not achieve very high performance and it relied mostly on NEs and country-specific expressions rather than on lexical or orthographic variation. MO and PT texts use the same orthography which is different from that used by BR texts. The assumption that newspapers texts from MO and PT are very similar was confirmed and we would like to investigate this in future work using other sets of features.

4.2 Visualization - Cluster Dendograms

To test the validity of the results obtained using supervised classification methods, we apply hierarchical clustering to obtain dendograms of the three language varieties. For this purpose we used the top 100 overused unigrams in MO articles (without NE removal) which in the supervised setup were the most helpful features for the MO class in distinguishing between BR, MO, and PT texts.

We first consider all texts from the same language variety as one cohesive dataset and merge them into one document, thus obtaining three large documents corresponding to

Rank	BR Orig.	BR No-NE	MO Orig.	MO No-NE	PT Orig.	PT No-NE
1	equipe	equipe	Macau	patacas	concelho	concelho
2	prefeito	prefeito	patacas	território	euros	euros
3	projeto	time	território	chinesa	freguesia	freguesia
4	fato	fato	Serviços	actriz	autarquia	autarquia
5	time	projeto	actriz	chinês	Leiria	portagens
6	você	você	China	residentes	Aveiro	distrital
7	diretor	diretor	chinesa	casinos	Braga	âmbito
8	ações	equipes	Chan	macaense	Algarve	autarcas
9	atividades	gol	Kong	territórios	Minho	orçamento
10	atual	ação	Território	casino	Novas	algarvia

Table 6: Overuse of lexical features in BR, MO, and PT texts

our three sub-corpora. Figure 2 shows the dendogram when hierarchical clustering with the average method and Euclidian distance is applied to the three merged sub-corpora.[7]

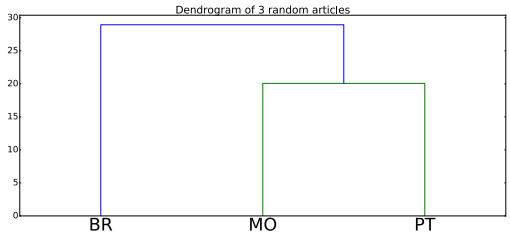


Figure 2: Dendogram of articles merged by language variety

In this simple dendogram we can see that the MO and PT datasets are displayed in the same branch of the dendogram whereas the BR dataset stands out. To confirm this, we subsequently carried out hierarchical clustering in five iterations over 16 randomly sampled articles from each sub-corpora (48 documents in total).[8] Figure 3 shows one of these iterations. MO articles are generally clustered much faster and closer to PT articles than to BR ones.

The use of hierarchical clustering confirms the results obtained using supervised text classification and indicates that the language used in articles published in Macau is substantially more similar to the language used in Portuguese newspapers than that used in Brazilian newspapers.

5 Conclusion

In this paper we proposed a supervised text classification approach to the study of language variation in Portuguese newspapers. Along with text classification we carried out a concise yet informative linguistic analysis of the most informative features in classification and we experimented with hierarchical clustering and dendograms to confirm the results obtained in the text classification experiments.

We focused on journalistic texts published in Macau in comparison to those published in Brazil and Portugal. We used short excerpts of texts available in the DSL Corpus Collection (DSLCC) version 2.1 [Tan *et al.*, 2014]. Our results confirmed our initial hypothesis that the language used in Portuguese newspapers published in Macau is much more similar to the language used in texts published in Portugal than to the one used in Brazilian newspapers.

We provided quantitative and qualitative evidence that journalistic texts published in Brazil and Macau differ substantially from each other. Our SVM classifier using a unigram language model can discriminate between texts from these two countries with 93% accuracy. Our results indicate that the same is not true for texts from Portugal and Macau. Texts from these two corpora could not be easily distinguished from each other by the SVM classifier. The analysis we carried out on the most informative features indicate that the main differences between texts published in Macau and Portugal captured by the classifier are country-specific expressions such as place names, currency name, etc.

5.1 Future Work

Our paper is, to the best of our knowledge, the first attempt to study the language of Portuguese newspapers published in Macau using NLP methods. We would like to test other features in future work such as word trigrams, POS tags and other forms of delexicalized text representations [Lui *et al.*, 2014] to investigate whether there are specific grammatical constructions prominent in Macanese journalism that are not used so often in the other two Portuguese varieties. In future work we would also like to investigate variation in the style of texts published in the newspapers from these three countries. To this end we are using readability metrics such as sentence length and lexical density.

Finally, we would like to investigate whether native speakers of different Portuguese varieties are able to discriminate Macanese texts from Brazilian and European texts. [Ács *et al.*, 2015; Goutte *et al.*, 2016] report that classification algorithms are able to obtain higher performance than humans distinguishing between Brazilian and European texts. We would like to investigate if this is true for Macanese texts as well.

Acknowledgments

Liviu P. Dinu was supported by UEFISCDI, PNII-IDPCE-2011-3-0959.

[7] We used the linkage and dendogram methods in SciPy [Jones *et al.*, 2001] and matplotlib to obtain this image.

[8] The number of documents used in this step was defined to optimize the dendogram visualization.

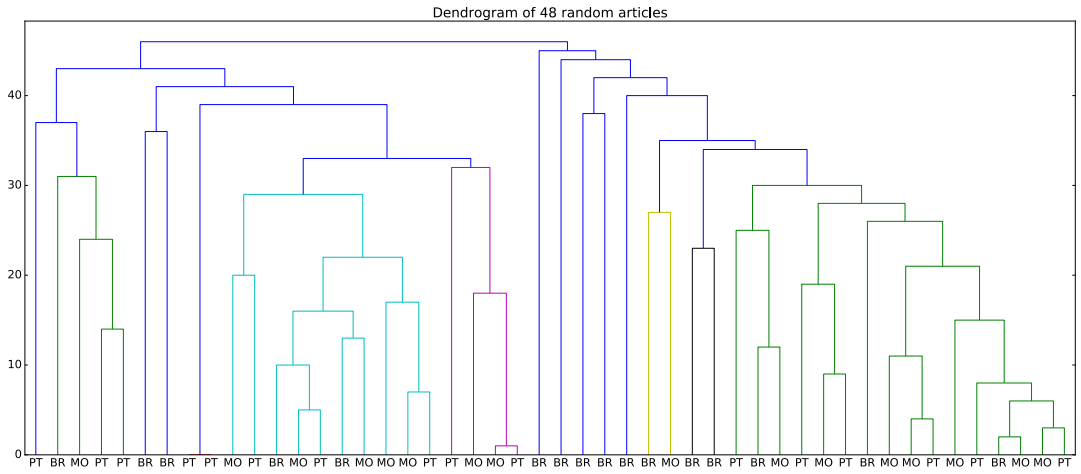


Figure 3: Hierarchical clustering of random sampled articles

References

[Ács *et al.*, 2015] Judit Ács, László Grad-Gyenge, and Thiago Bruno Rodrigues de Rezende Oliveira. A Two-level Classifier for Discriminating Similar Languages. In *Proceedings of LT4VarDial*, 2015.

[Amaro, 2016] Vanessa Amaro. Linguistic Practice, Power and Imagined Worlds: The Case of the Portuguese in Postcolonial Macau. *Journal of Intercultural Studies*, 37(1):33–50, 2016.

[Baxter, 1992] Alan N Baxter. Portuguese as a pluricentric language. *Pluricentric languages: differing norms in Different Nations*, (62):11, 1992.

[Baxter, 1996] Alan N Baxter. Portuguese and Creole Portuguese in the Pacific. *Atlas of languages of intercultural communication in the Pacific, Asia, and the Americas*, 3:299, 1996.

[Ciobanu and Dinu, 2016] Alina Maria Ciobanu and Liviu P. Dinu. A Computational Perspective on Romanian dialects. In *Proceedings of LREC*, 2016.

[Fan *et al.*, 2008] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[Goutte *et al.*, 2014] Cyril Goutte, Serge Léger, and Marine Carpuat. The NRC System for Discriminating Similar Languages. In *Proceedings of VarDial*, 2014.

[Goutte *et al.*, 2016] Cyril Goutte, Serge Léger, Shervin Malmasi, and Marcos Zampieri. Discriminating Similar Languages: Evaluations and Explorations. In *Proceedings of LREC*, 2016.

[Jones *et al.*, 2001] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001.

[Lui and Cook, 2013] Marco Lui and Paul Cook. Classifying English Documents by National Dialect. In *Proceedings of ALTA*, 2013.

[Lui *et al.*, 2014] Marco Lui, Ned Letcher, Oliver Adams, Long Duong, Paul Cook, and Timothy Baldwin. Exploring Methods and Resources for Discriminating Similar Languages. In *Proceedings VarDial*, 2014.

[Maier and Gómez-Rodrıguez, 2014] Wolfgang Maier and Carlos Gómez-Rodrıguez. Language Variety Identification in Spanish Tweets. In *Proceedings of LT4CloseLang*, 2014.

[Malmasi and Dras, 2015a] Shervin Malmasi and Mark Dras. Automatic Language Identification for Persian and Dari Texts. In *Proceedings of PACLING*, 2015.

[Malmasi and Dras, 2015b] Shervin Malmasi and Mark Dras. Language Identification Using Classifier Ensembles. In *Proceedings of LT4VarDial*, 2015.

[Malmasi *et al.*, 2015] Shervin Malmasi, Eshrag Refaee, and Mark Dras. Arabic Dialect Identification Using a Parallel Multidialectal Corpus. In *Proceedings of PACLING*, 2015.

[Tan *et al.*, 2014] Liling Tan, Marcos Zampieri, Nikola Ljubešić, and Jörg Tiedemann. Merging Comparable Data Sources for the Discrimination of Similar Languages: The DSL Corpus Collection. In *Proceedings of BUCC*, 2014.

[Zampieri and Gebre, 2012] Marcos Zampieri and Binyam Gebrekidan Gebre. Automatic Identification of Language Varieties: The Case of Portuguese. In *Proceedings of KONVENS*, 2012.

[Zampieri *et al.*, 2013] Marcos Zampieri, Binyam Gebrekidan Gebre, and Sascha Diwersy. N-Gram Language Models and POS Distribution for the Identification of Spanish Varieties. In *Proceedings of TALN*, 2013.

[Zampieri *et al.*, 2015] Marcos Zampieri, Liling Tan, Nikola Ljubešić, Jörg Tiedemann, and Preslav Nakov. Overview of the DSL Shared Task 2015. In *Proceedings of LT4VarDial*, 2015.

NLP-driven Data Journalism: Time-Aware Mining and Visualization of International Alliances

Xavier Tannier
LIMSI, CNRS, Univ. Paris-Sud,
Université Paris-Saclay
F-91405 Orsay, FRANCE
xavier.tannier@limsi.fr

Abstract

We take inspiration of computational and data journalism, and propose to combine techniques from information extraction, information aggregation and visualization to build a tool identifying the evolution of alliance and opposition relations between countries, on specific topics. These relations are aggregated into numerical data that are visualized by time-series plots or dynamic graphs.

1 Introduction

Information and communication technologies have provided tools and methods to make the production of information more democratic. As a result, a vast amount of content is available, which arguably creates more noise than knowledge at the end of the day. Without hierarchical organization and contextualization, users may lack perspective to understand and assimilate the multiplicity of events that they come across every day, and to link them to related events in the past.

In this context, journalists and technologists developed the notion of "data journalism", which takes advantage of the growing popularity of Open Data, the development of structured knowledge bases such as DBPedia [Lehmann *et al.*, 2013], YAGO [Hoffart *et al.*, 2013] or OpenCalais and many others, as well as recent work in data visualization, to facilitate information analysis and access a variety of points of view. However, knowledge is still far from being entirely represented in structured databases, and much information remains available in text format. For this reason, natural language processing has a lot to offer to modern journalism.

In this paper, we present a tool that automatically identifies alliance and opposition relations between countries, on a specific subject defined by a user query (*e.g.* situation in Syria, nuclear proliferation, North Pole ownership). The evolution of the relations over time are then illustrated by a time-series plot (see Figure 3) or dynamic graphs and maps (Figure 4).

2 Related Work

The idea of automatically finding topically related material in streams of newswire data goes back to Topic Detection and Tracking evaluation campaigns [Allan, 2002].

Our work contributes to this research effort and explores the quantitative aspects of knowledge that can be extracted from textual documents. With that respect, as well as on the topic of alliance and opposition relations, this can be connected to opinion mining. [Chambers *et al.*, 2015] also consider these relations, based on Twitter data.

We also use well-known techniques of feature-based, supervised relation extraction. The aim is to identify and classify relations between two entities in the same sentence, by learning these relations on a training, manually annotated dataset. Examples of such works are [Miller *et al.*, 2000], [Kambhatla, 2004], [Boschee *et al.*, 2005], [Zhou *et al.*, 2005], among many others. Our approach differs from most works in the fact that each relation is associated to a date, which makes the classification time-aware.

Also, we bias our classifier towards precision. This approach relies on linguistic variation and redundancy in a large amount of documents to ensure a good coverage. It is related to works in question-answering [Dumais *et al.*, 2002], temporal information aggregation [Kessler *et al.*, 2012] or opinion mining [Turney, 2002].

3 System Overview

We apply information extraction techniques to a large amount of newswire textual documents, in order to acquire enough data to make significant statistics on them. These data can then be accessed by a query-based visualization tool.

Relations that we extract are opposition (*NEG*) or alliance (*POS*) relations between two countries, explicitly expressed in a same sentence, such as:

(1) **Indonesia** voiced support for **East Timor**'s bid to join the ASEAN. → *POS(Indonesia, East Timor)*

(2) **London**'s recent condemnations of **Libyan leader Moamer Kadhafi**'s bloody crackdown [...]. → *NEG(U.K., Libya)*

(3) **Chavez** has stoop up for his longtime ally **Kadhafi**, [...]. → *POS(Venezuela, Libya)*

The alliance or opposition can be made explicit mainly by an action verb (*protested*), an event noun (*condemnations*) or a state noun (*ally*). Countries (relation arguments) can be designated by their actual names, the name of their capital or of a person representing this country.

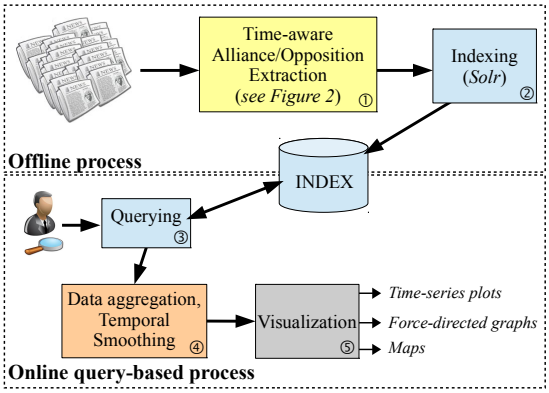


Figure 1: System overview.

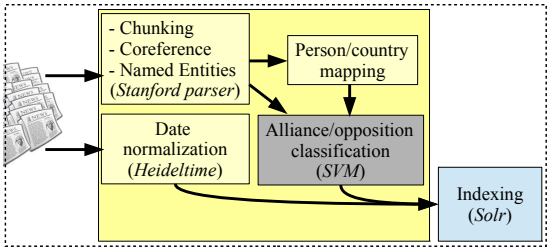


Figure 2: Offline processing details.

Figure 1 shows the general architecture of the system. At indexing time, the corpus is processed (step ① and Figure 2) with chunking, coreference resolution, named entity tagging, temporal information normalization and association between some people names and their countries. A classifier extracts *POS* and *NEG* relations in each sentence.

At query time, all sentences that are relevant to the query are retrieved (③). For each day, *POS* and *NEG* relations are aggregated and a tendency of this day is obtained, as well as a trend over time (④). A graphical visualization of this trend is then proposed to the user (⑤ and Figures 3, 4).

4 Alliance and Opposition Classification

4.1 Resources

We used a corpus of English newswire texts provided by the AFP French news agency. The English AFP corpus is composed of 1.79 million texts that span the 2004-2013 period (635 documents/day in average and 600 millions words).

Entities. Country, nationality and capital name lists were extracted from the linked data repository of the CIA World Factbook, and linguistic variations of these names were collected from DBPedia. All these elements were considered as entities potentially parts of an alliance/opposition relation.

Alliance/Opposition Gazetteers. We constituted manually a restrained lexicon of "relation triggers" containing 110 words used in the corpus to express alliance and opposition (and not a more general "polarity"). These words are verbs (*agree*, *support*, *accuse*, *condemn*, *slam*...), event nouns (*congratulation*, *accusation*, *sanction*...) and other nouns (*ally*...).

Date Normalization. Our aim is to associate alliance and opposition relations to dates, in order to observe the evolution of these relations in time. We used Heideltime [Strötgen and Gertz, 2013] to normalize dates inside the documents. Normalization is the operation of turning a temporal expression (*e.g.*, *July 26th*, *yesterday*, *Last Tuesday*) into a formated, fully specified representation (*2013-07-26*).

Chunking. Parse trees are obtained from the Stanford parser [Klein and Manning, 2003], and we extract minimal chunks (NPs, VPs) from these trees.

4.2 Classification

It is important to note that we do not claim to build a generic classifier of such relations. Our aim is to extract enough relations to be able to produce realistic, substantial and significant data about international relations between countries. At classifier level, a good precision will be favored, because we rely on the redundancy of information in the collection to achieve an appropriate coverage (*i.e.*, recall) of relations. Therefore, we will learn a classifier to identify relations that are expressed explicitely, in the same sentence.

Training Set. We randomly selected sentences containing at least two entities separated by less than 15 chunks, and at least one relation trigger between the two entities.

Each pair of entities satisfying these constraints are an instance of the classifier, which means that one sentence can lead to several instances. *e.g.*, in:

(4) In Jerusalem, Prime Minister **Silvio Berlusconi** pledged **Italy**'s firm *support* for **Israel** and urged effective *sanctions* against **Tehran**.

Entities are in bold and relation triggers in italics. Relations to classify are then {Italy (S. Berlusconi), Israel}, {Italy (S. Berlusconi), Iran (Tehran)}, {Italy, Israel}, {Italy, Iran (Tehran)} and {Israel, Iran (Tehran)}. The resulting relations would be *NEG(Italy, Iran)* and *POS(Italy, Iran)*.

We annotated 2105 such instances with relation NIL (no relation, 1463 instances), NEG (opposition, 349 instances) and POS (alliance, 293 instances).

Classifier. Sentences do not differ in their structure whether they express alliance or opposition. Entities have the same kinds of interactions with each other. Only the polarity of trigger words matters. Therefore, we implement a two-step SVM classification:

A. Filtering out pairs that have no relation at all, *i.e.* classifying between *NIL* and *non-NIL* relations;

B. Among non-NIL relations, classifying between *POS* and *NEG* relations;

These classifiers were evaluated by a 10-fold cross-validation. Results are presented in Table 2. The final model used for next steps is trained with all annotated instances.

5 Time-Aware Aggregation and Visualization

The alliance classifier described in previous section extracts a total of 330,222 instances of relations from the corpus. If a date has been identified and normalized in a sentence, then this date stamps the sentence. If Heideltime was unable to

STEP A
Entities
- Distance between entities, sentence length, positions of entities - Type of E_1 and E_2 (capital, person name, country name) - Number of entities around and inside entities
Lexical features
- Presence, number and type (verb, noun) of triggers around or between entities. - Negation between E_1 and E_2 - Among 20 most frequent prepositions in the corpus: those occurring just before and just after E_1 or E_2
Syntactic features
- Whether E_1 (resp. E_2) is the head of its chunk, size of chunks - Number of verbs, nouns around entities

STEP B
Distinction between NEG and POS triggers, negation
- Number of positive (resp. negative) triggers in the sentence - Number of positive (resp. negative) triggers around entities - Negation marks

Table 1: Features used for the classifiers A and B.

	Relation	Precision	Recall	F1
Step A	**non-NIL**	**0.82**	0.76	0.79
	NIL	0.90	0.93	0.91
	average	0.87	0.88	0.87
Step B	POS	0.95	0.99	0.97
	NEG	0.99	0.95	0.97
	average	**0.97**	**0.97**	0.97
A + B	**POS & NEG**	**0.80**	0.73	0.76

Table 2: 10-fold cross validation results for the alliance/opposition classifier.

normalize the date, then the sentence is skipped. Otherwise (no date found), the document creation time stamps the sentence. A typical query is then composed of a few keywords representing the topic, a temporal interval (minimum of maximum dates) and zero or more country names on which the user wants to restrict relation extraction.

For all pairs of considered countries, inside the same day d, the weight for the pair and the day d is:

$$w(d) = log(\frac{1 + P(d)}{1 + N(d)})$$

where $P(d)$ and $N(d)$ are the number of *POS* and *NEG* relations between the two countries. $w(d)$ is a number between $-\infty$ and $+\infty$, where $w = 0$ is neutral, $w < 0$ is an opposition and $w > 0$ is an alliance. The noise is then reduced by a weighted mean smoothing over a temporal window of 5 days.

5.1 Visualization

For bilateral relations (field *countries* containing two items), we provide the user with a time-series plot representing $sw(d)$, and show them on demand the sentences which led to this value. Figure 3 shows an example of results concerning the relations between United States and Russia ("*countries:United States AND Russia*") concerning the situation in Syria ("*keywords:syria*").

When zero, one or more than two countries are specified by the user, we generate a graph of countries, where the distance between vertices reflects the opposition between the countries in a given time span (Figure 4). We use the Barnes-Hut force-directed layout algorithm, where a value between two vertices is considered as a repulsive force. The color of the nodes reflects the their proximity with each other (using a first-neighbor shortest path algorithm).

For this algorithm, we need to transform our weights $sw(d)$ into positive numbers (repulsive forces). We also need to damp the noise and the variations of the weights that are introduced by the volume of data. For example, two positive values of 1 and 3.5 (diff. = 2.5) should be considered as close to each other, while a positive value of 1 should be far from a negative value of -1 (diff. = 2). This kind of effects can be corrected by a "S-shaped", logistic function, that models a level of saturation after an approximately exponential growth (or, in our case, decrease):

$$sw'(d) = 1 - \frac{1}{1 + e^{-sw(d)}} \quad (5)$$

This function levels off high weights (both negative and positive), increases differences between positive and negative values and thus helps reducing noise without having to discretize values arbitrarily. Resulting numbers are all positive, between 0 and 1, where 1 is a strong opposition (then, repulsion in the graph) and 0 is a strong alliance (attraction in the graph), while 0.5 is neutral.

5.2 Evaluation

Evaluating the relevance of the produced trends is very subjective and would require a high level of expertise in every tested domain. Even if this work is carried out in collaboration with journalists, we cannot afford such an effort. That is why we opted for a protocol that is at the same time more objective and easier to conduct:

1. We chose 14 queries with the following information: names of two countries (or unions of countries) involved in the relation, and an optional keyword-based thematic restriction. We selected queries having potentially a high density of extracted relations (*e.g.* North Korea *vs.* South Korea, or Russia *vs.* United Nations on "Syria"), as well as sparser topics (France *vs.* Germany on "austerity", China *vs.* Japan on maritime affairs).
2. On the resulting plot, we selected up to 5 strong peaks — $abs(sw(d)) > 1$ — and 5 weak peaks — $abs(sw(d)) \leq 1$ (total of 97 relations).
3. For each of these peaks, we estimated whether the polarity of the peak was relevant or not. For that purpose, we seeked news articles from a time-stamped web collection that was not part of the tested collection, in order to validate whether the two countries rather agreed or opposed at the date indicated by the peak. This is still a heavy task, which explains the low number of tested instances.

The accuracy of strong peaks is 0.90, making them highly reliable. Accuracy of weak peaks is 0.702. Note that very

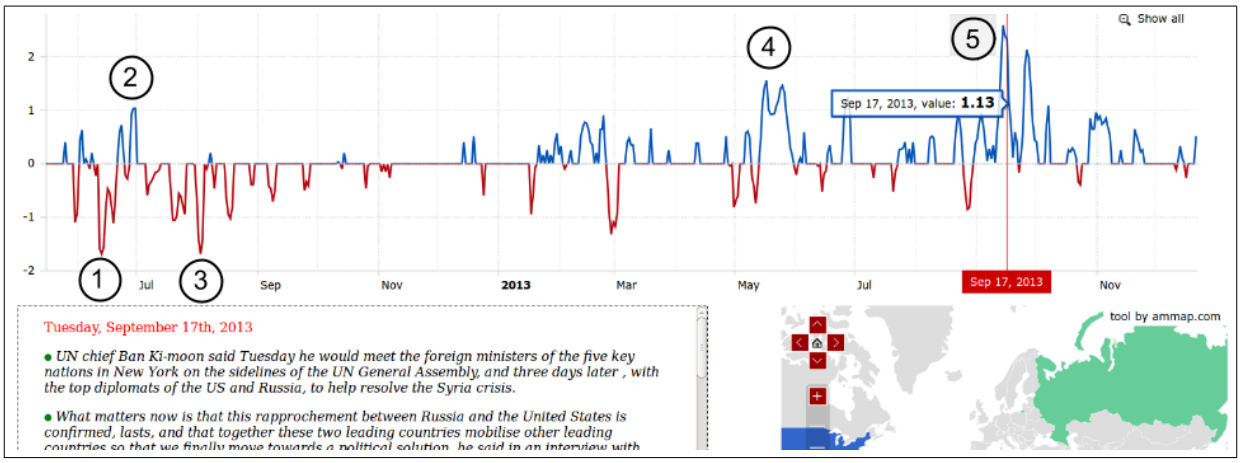


Figure 3: Example of plot produced by the system for bilateral relations between United States and Russia on the query "Syria". The bottom left frame shows sentences corresponding to the user-selected date (Sep. 17, 2013). Circled numbers have been manually added to the screenshot. They correspond to: ① Mutual accusations of supplying arms to Syrian authorities or opposition (bad relation, $sw(d) << 0$); ② Planning of a meeting to discuss the problem (better relation, $sw(d) > 0$); ③ Vetos of China and Russia for United Nations resolutions; ④ Announcement of a peace conference; ⑤ Agreement at this conference.

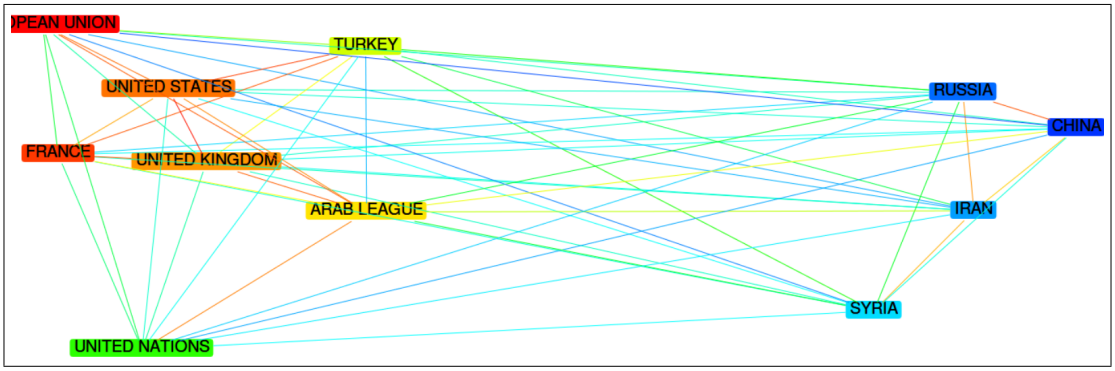


Figure 4: Example of graph produced by the system for relations between different states on the query "syria", for the year 2012. The graph is based on information collected in 18,582 sentences. Edge colors indicate the kind of relation (from dark red for strong alliance to dark blue for strong opposition), and vertice colors reflects proximity of countries with each other.

small peaks (< 0.4) are far less reliable. A better data smoothing could reduce this problem.

6 Discussion and Perspectives

The system described in this paper shows that it is possible to use NLP techniques to aggregate information and provide reliable numerical data that could hardly be obtained in another way. This kind of NLP-driven data journalism can bring significant added value to journalists, as well as end users, in many fields. This opens the way for many applications of the same kind, for studying relations between people, countries, organisations in any domain.

The main guidelines for building such a system are:

- Think about precision first. Do not neglect recall, since it is all about getting data, but the vast amount of information can make up the lack of coverage.
- High variation in the corpus is better than low variation, to have more chance to get data from your accurate classifier, and to give less importance to misclassifications.
- Smoothing the resulting data within temporal windows and correcting them with logistic functions is essential for hiding errors and producing reliable information.
- Make it time-aware. Interests are two-fold: some data are valid only in a limited time range, and the evolution of data can be a main interest of the study.

A wide new range of knowledge can thus become available to data journalists, who generally make use of factual data from structured bases. Such applications would bring high added value to the final users, in terms of aggregation, contextualization and hierarchical organization of information. We need however to reduce drastically the amount of needed supervision in order to make NLP enter the journalism world.

References

[Allan, 2002] J. Allan, editor. *Topic Detection and Tracking*. Springer, 2002.

[Boschee *et al.*, 2005] E. Boschee, R. Weischedel, and A. Zamanian. Automatic information extraction. In *Proceedings of the International Conference on Intelligence Analysis*, 2005.

[Chambers *et al.*, 2015] N. Chambers, V. Bowen, E. Genco, X. Tian, E. Young, G. Harihara, and E. Yang. Identifying Political Sentiment between Nation States with Social Media. In *Proceedings of EMNLP 2015*.

[Dumais *et al.*, 2002] S. Dumais, M. Banko, E. Brill, J. Lin, and A. Ng. Web Question Answering: Is More Always Better? In *Proceedings of the 25th Annual International ACM SIGIR Conference*.

[Hoffart *et al.*, 2013] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia. *Artificial Intelligence*, 2013.

[Kambhatla, 2004] N. Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *Proceedings of the 42nd Annual Meeting of the ACL*, 2004.

[Kessler *et al.*, 2012] R/ Kessler, X. Tannier, C. Hagège, V. Moriceau, and A. Bittar. Finding Salient Dates for Building Thematic Timelines. In *Proceedings of the 50th Annual Meeting of the ACL*, 2012.

[Klein and Manning, 2003] Dan Klein and Christopher D. Manning. Accurate Unlexicalized Parsing. In *Proceedings of the 41st Meeting of the ACL*, 2003.

[Lehmann *et al.*, 2013] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer. DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia. *Semantic Web Journal*, 2013.

[Miller *et al.*, 2000] S. Miller, H. Fox, L. Ramshaw, and R. Weischedel. A novel use of statistical parsing to extract information from text. In *Proceedings of NAACL*, 2000.

[Strötgen and Gertz, 2013] J. Strötgen and M. Gertz. Multilingual and Cross-domain Temporal Tagging. *Language Resources and Evaluation*, 47(2):269–298, 2013.

[Turney, 2002] P. D. Turney. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. In *40th Annual Meeting of the ACL*, 2002.

[Zhou *et al.*, 2005] G. Zhou, J. Su, J. Zhang, and M. Zhang. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd Annual Meeting of the ACL*, 2005.

Semantic and Context-aware Linguistic Model for Bias Detection

Sicong Kuang **Brian D. Davison**
Lehigh University, Bethlehem PA
sik211@lehigh.edu, davison@cse.lehigh.edu

Abstract

Prior work on bias detection has predominantly relied on pre-compiled word lists. However, the effectiveness of pre-compiled word lists is challenged when the detection of bias not only depends on the word itself but also depends on the context in which the word resides. In this work, we train neural language models to generate vector space representation to capture the semantic and contextual information of the words as features in bias detection. We also use word vector representations produced by the GloVe algorithm as semantic features. We feed the semantic and contextual features to train a linguistic model for bias detection. We evaluate the linguistic model's performance on a Wikipedia-derived bias detection dataset and on a focused set of ambiguous terms. Our results show a relative F1 score improvement of up to 26.5% versus an existing approach, and a relative F1 score improvement of up to 14.7% on ambiguous terms.

1 Introduction

Bias in reference works affects people's thoughts [Noam, 2008]. It is the editor's job to correct those biased points of view and keep the reference work as neutral as possible. But when the bias is subtle or appears in a large corpus, it is worth building computational models for automatic detection. Most prior work on bias detection rely on pre-compiled word lists [Recasens *et al.*, 2013; Iyyer *et al.*, 2014; Yano *et al.*, 2010]. This approach is good at detecting simple biases that depend merely on the word. Such methods are appropriate when the word itself indicates strong subjectivity polarity or the author's stance intuitively and straightforwardly. In Examples 1a and 2a shown below[1], both "terribly" and "disastrous" are subjective words indicating the author's negative emotion; the word "terrorist" in Example 3a clearly identifies the author's stance on the event. Use of a pre-compiled word list is sufficient to detect such words.

1. (a) The series started **terribly** for the Red Sox.
   (b) The series started very **poorly** for the Red Sox.

[1] All examples in this work are extracted from the dataset derived from Wikipedia 2013 [Recasens *et al.*, 2013].

2. (a) Several notable allegations of lip-synching have been recently targeted at her due to her **disastrous** performances on Saturday Night Live.
   (b) Several notable allegations of lip-synching have been recently targeted at her due to her **poor** performances on Saturday Night Live.

3. (a) **Terrorists** threw hand grenades and opened fire on a crowd at a wedding in the farming community of Patish, in the Negev.
   (b) **Gunmen** threw hand grenades and opened fire on a crowd at a wedding in the farming community of Patish, in the Negev.

However, using a pre-compiled word list also has significant drawbacks. It is inflexible in the sense that only words appearing in the list can be detected. Words with similar meanings but not collected in the list would not be detected. Thus this method only focuses on the surface form of the word while neglecting its semantic meaning. Focusing on the word itself also means neglecting the context in which the word resides. But some bias can only be detected when contextual information is considered. Words associated with this kind of bias, such as "white" in Example 4a, are often ambiguous and hard to detect using only a pre-compiled word list. The meaning of such words can only be clarified by interpreting the context of the word. The modified sentence in each example is the correct version supplied by Wikipedia editors.

4. (a) By bidding up the price of housing, many **white** neighborhoods again effectively shut out blacks, because blacks are unwilling, or unable, to pay the premium to buy entry into white neighborhoods.
   (b) By bidding up the price of housing, many **more expensive** neighborhoods again effectively shut out blacks, because blacks are unwilling, or unable, to pay the premium to buy entry into white neighborhoods.

Recent years have seen progress in learning vector space representations for both words and variable-length paragraphs [Pennington *et al.*, 2014; Mikolov *et al.*, 2013b; Le and Mikolov, 2014a; Mikolov *et al.*, 2013a]. In this work, we use and build models to generate semantic and contextual vector space representations. Equipped with semantic and contextual information, we then build a semantic and context-aware linguistic model for bias detection.

2 Background

Current research in bias detection often uses both pre-compiled word lists and machine learning algorithms [Recasens *et al.*, 2013; Iyyer *et al.*, 2014; Yano *et al.*, 2010]. Most define the bias detection problem as a binary classification problem. Gentzkow and Shapiro [2010] select 1,000 phrases based on the frequency that these phrases appear in the text of the 2005 *Congressional Record*. They form a political word list that can separate Republican representatives from Democratic representatives as the initial step in detecting the political leaning of the media. Greenstein and Zhu [2012] applied Gentzkow and Shapiro's method to Wikipedia articles to estimate Wikipedia's political bias. Their result shows many Wikipedia articles contain political bias and the polarity of the bias evolves over time.

Sentiment analysis in bias detection is often used to detect a negative tone or a positive tone of a sentence or a document which should have been neutral [Kahn *et al.*, 2007; Saif *et al.*, 2012]. This kind of bias in reference works is easier to detect due to the emotional identifier it uses, usually an adjective. Recasens et al. [2013] use a pre-compiled word list from Liu et al. [2005] to detect non-neutral tone in reference works. Yano et al. [2010] evaluated the feasibility of automatically detecting such biases using Pennebaker et al.'s LIWC dictionary [2015] compared to human judgments using Amazon Mechanical Turk in the politics domain.

We learn word and document vector representations from two neural language models [Le and Mikolov, 2014b] and GloVe algorithms [Pennington *et al.*, 2014]. The word vectors and document vectors are used as semantic and contextual features to build a linguistic model. Below we introduce the models and algorithm we use to learn the features.

Neural Language Model

Neural language models are trained using neural networks to obtain vector space representations [Bengio *et al.*, 2006]. Although the vector space representations of the words in a neural language model are initialized randomly, they will eventually learn the semantic meaning of the words through the prediction task of the next word in a sentence. [Mikolov *et al.*, 2013b; Le and Mikolov, 2014b]. Using the same idea, we treat every document also as an unique vector. And the document vector will eventually learn the semantics through the same prediction task as we do for word vector.

We use stochastic gradient descent optimization algorithm via backpropagation algorithm to train document vector representations and word vector representations. The model that considers the document vector as the topic of the document or the contextual information when predicting the next word, is called the Distributed Memory Model (dm). Since in the process of building a dm model, word vectors in the corpus will capture the semantic meanings; in our work, besides using the dm model to learn document vectors as contextual features, we also use the dm model to learn word vectors as semantic features. The Distributed Bag of Words model (dbow) only learns document vector representations and it is trained by predicting words randomly sampled from the document [Le and Mikolov, 2014b]. In this work, we also use dbow model to learn document vectors as contextual features.

GloVe Algorithm

In both the dm and dbow models, text is trained from a local context window. By utilizing global word-word co-occurrence counts, the ratio of co-occurrence probabilities are able to capture the relevance between words. Pennington et al. [2014] use this idea to construct a word-word co-occurence matrix, and reduce the dimensionality by factorization. The resulting matrix contains vector space representations for each word. In this work, we use GloVe's pre-trained word vectors learned from Wikipedia in 2014[2] as semantic features to train a linguistic model.

3 Approach

Our work extends the work of Recasens et al. [2013], who use eight pre-compiled word lists to generate boolean features to train a logistic regression model to detect biased words. In Recasens's work, 32 manually crafted features for each word being considered are utilized to build a logistic regression model. Among the features, about two thirds of their features (20/32) are boolean features derived from the pre-compiled word lists. Other features include the word itself, lemma, part of speech (POS) and grammatical relation.

By using pre-compiled word lists, their method neglects semantic and contextual information. Moreover, in their evaluation, they evaluate their model's performance as the ratio of sentences with the correctly predicted biased word. This metric has two flaws: first using a word-feature matrix as input, the linguistic model is a word-based classification model and thus word-based evaluation metrics are needed; second, to calculate the sentence-based metric, the authors obtain the predicted probabilities for all words in the sentence—the word with the highest probability is predicted as the biased word. The authors' implicit assumption is that there must exist a biased word in every sentence, which is not the case in real-world text. Since the dataset is derived from Wikipedia, non-biased words form the majority class and so accuracy is not an effective metric. In contrast, we focus on the model's quality on detection of biased words. To address the above problems, we use word-based evaluation metrics—precision, recall and F1 score—to evaluate performance.

In this work, we train two neural language models using stochastic gradient descent and backpropagation, a distributed memory model and a distributed bag of words model, to learn vector space representations to capture the contextual information of each word under consideration. Our assumption is that equipped with contextual information the linguistic model should be better able to detect bias associated with ambiguous words. To tackle the problem that the pre-compiled word list method only focuses on remembering the form of the words in the list, we use recent approaches from Pennington et al. [2014] and Mikolov et al. [2013a; 2014b] to obtain vector space representations that can capture the fine-grained semantic regularities of the word. We incorporate the semantic features and contextual features when building a logistic regression model for the bias detection task.

[2]http://nlp.stanford.edu/projects/glove/

4 Experiment and Analysis

Since our task comes from Recasens et al. [2013], we aim to build a linguistic model to detect framing bias and epistemological bias. Recasens et al. used multiple boolean features derived from pre-compiled word lists (true if in the list, false otherwise) to describe the target word. Our first expectation is that by using the finer structure of the word vector space using methods by Pennington et al. [2014] and Mikolov et al. [2013a], the finer-grained semantic regularities should become more visible and thus get better bias detection performance because similar words will be classified similarly. Second, by generating document vector space representations to capture the context of each word, we should improve the model's performance on bias detection associated with ambiguous words, since we can potentially distinguish different uses of the same word.

We use Recasens et al.'s approach as baseline. To better understand the behavior of the semantic features and the contextual features, we design our experiments to be in three scenarios: first we retain all the features in Recasens et al.'s work and only add our semantic features to train a logistic regression model; second we retain all the features in Recasens et al.'s work and add our contextual features to train a logistic regression model; third we add both the semantic and contextual features. In their work, Recasens et al.'s feature space consists (in part) of lexical features (word and POS) and syntactic features (grammatical relationships). A list of all 32 features may be found in Recasens et al. [2013].

To better measure the contextual feature's behavior in detecting bias associated with ambiguous words, we extract a focused subset of the test cases consisting of ambiguous words (i.e., those in the training set that are inconsistently labeled as biased). We measure the precision, recall and F1 score of the focused set before and after we add the contextual features. The logistic regression model computes each word's probability to be biased. We derive a threshold probability to decide beyond which the words should be predicted as biased by choosing the threshold when the F1 score is maximized on the training set, examining thresholds across (0, 1) using intervals of 0.001.

4.1 Dataset

Wikipedia endeavors to enforce a neutral point of view (NPOV) policy[3]. Any violation of this policy in the Wikipedia content will be corrected by Wikipedia editors. As a free online reference, Wikipedia publishes its data dumps once per month (English version Wikipedia). By doing a *diff* operation on the same Wikipedia articles from two different Wikipedia dumps, we are able to extract the "before form" string (the sentence with a single biased word from the old Wikipedia article) and the "after form" string (the same sentence with the biased word corrected by the Wikipedia editors) [2013]. With such a labeled data set from Wikipedia, we are ready to build a linguistic model to automatically detect biased words in a reference work.

We use the raw datset from Recasens et al. [2013] derived from articles from Wikipedia in 2013. The biased words are labeled by Wikipedia editors. However, since some details of their data preparation are not included in their paper, our statistics of the dataset after processing and cleaning (shown in Table 1) are slightly different from theirs.

[3]https://en.wikipedia.org/wiki/Wikipedia:Neutral_point_of_view

Data	Number of sentences	Number of words
Train	1779	28638
Test	207	3249
Focused set	NA	706

Table 1: Statistics of the dataset

	baseline	dm doc vec	dbow doc vec	dm doc vec + dbow doc vec
# features	32	332	332	632
precision	0.245	0.228	0.228	0.224
recall	0.228	0.335	0.335	0.330
F1 score	0.236	0.271	0.271	0.267

Table 2: Results on test set after adding contextual features

4.2 Baseline

For our baseline, we built a logistic regression model using the approach of Recasens et al. [2013]. To better prepare the data, we also added the following steps in data cleaning which are not specified in their paper: we discard data tuples in both training set and test set if the "before form" string and "after form" string only differ by numbers or contents inside $\langle\rangle$ and $\{\}$, since contents inside $\langle\rangle$ and $\{\}$ are not text in Wikipedia and we also ignore the words within $\langle\rangle$ and $\{\}$ when we generate the word-feature matrix. We also remove tuples from the dataset in which the biased word belongs to the stopwords set. Moreover, we use regex to check and remove those tuples if the biased word of that tuple happens to be in the Wikipedia article's title. We use the Stanford CoreNLP (version 3.4.1) [Marneffe *et al.*, 2006] to generate grammatical features, such as part of speech, lemma and grammatical relationships. The result of the baseline is shown in the first column of Table 2.

4.3 Experiment on Contextual Features

For each word in the data set, we generate fixed length vector representations of the Wikipedia articles in which the word resides as the contextual features by training two neural language models. This fixed length document vector of the article, together with the original 32 features from Recasens et al.'s paper [Recasens *et al.*, 2013] will be the input to train a logistic regression model to perform bias detection.

To generate the contextual features for each word in the dataset, we use all 7,464 Wikipedia articles and altogether 1.76 million words as input to train two neural language models, a distributed memory model (dm) and a distributed bag of words model (dbow), using the open source package gensim on a 128GB memory machine with 16 3.3 Ghz cores. The training process took approximately 5 hours using 16 workers (cores). For each model, we iterate over 10 epochs. For each Wikipedia article, we split and clean it using the same procedures as we process the *"before form"* strings [Recasens *et al.*, 2013]. For each article, we use the Wikipedia article name as the label to train the neural language model. For both models, we use a window size of 10 and vector dimension of 300

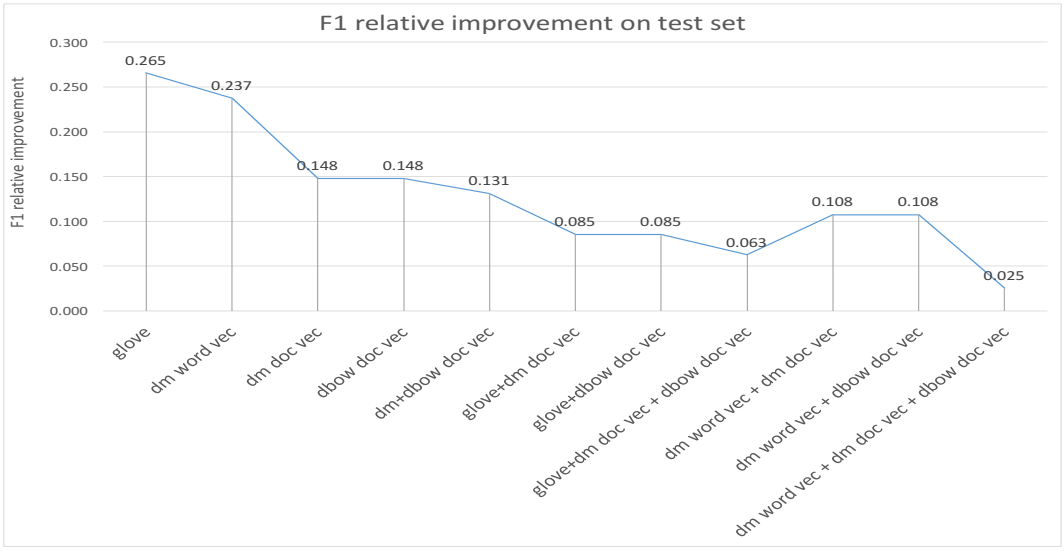


Figure 1: F1 relative improvement on test set

for the vector representations. As suggested by Mikolov and Le [2013b], we also experiment on the combination of dm and dbow vectors as contextual features.

For metrics, precision is defined as

$$\frac{\text{\# words predicted to be biased and labeled as biased}}{\text{\# words predicted to be biased}} \quad (1)$$

Recall is defined as

$$\frac{\text{\# words predicted to be biased and labeled as biased}}{\text{\# words labeled as biased}} \quad (2)$$

F1 score is defined as the harmonic mean of precision and recall

$$\frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (3)$$

We use F1 score to measure the overall performance of the linguistic model of the baseline. The result is shown in Table 2. We can see a decrease in the precision and an increase in the recall, which result in an overall increase of F1. This indicates a significant rise in false positives. Compared to the baseline, the precision of the contextual-aware model slightly drops. But we should point out that contextual features are only helpful when detecting bias associated with ambiguous words. There are relatively few ambiguous words (706 out of 3249) in the test set. For non-ambiguous words, the contextual features are not helping but increase the feature dimensionality.

4.4 Experiment on Semantic Features

To capture fine-grained semantic regularities of words, we use pre-trained word vectors of size 300 from the GloVe algorithm [Pennington *et al.*, 2014] trained on articles from Wikipedia 2014. Since the dm model can also learn the word vector representation inside its input documents, we also use the dm model to generate word vectors of size 300 as semantic features. The learned semantic features are used as input

	baseline	GloVe	dm word vec
# features	32	332	332
precision	0.245	0.284	0.304
recall	0.228	0.316	0.282
F1 score	0.236	0.299	0.292

Table 3: Results on test set after adding semantic features

to train a logistic regression model to classify bias, with the result presented in Table 3. The result shows that compared to contextual features, semantic features generally performs better in this task. Semantic features trained by the GloVe algorithm give the best F1 score. This suggests that semantic features trained either by GloVe or the dm model could significantly improve a linguistic model's performance on bias detection.

4.5 Combination of Semantic and Contextual Features

To see if the two types of features together can strengthen the logistic regression model's power in detecting bias, we try different combinations of semantic and contextual features to build linguistic models. The relative improvement of F1 score of different combinations against baseline is shown in Figure 1. The result shows in general semantic features alone perform better than both contextual features and the combinations of those two. The result shows by adding the GloVe as semantic features alone can reach a relative improvement of up to 26.5%. The group of results after adding contextual features alone gives second tier best result showing the model can learn from contextual features along. However, the performance drop significantly when combining semantic and contextual features. After adding contextual features, the relative ratio of F1 drops. However, we cannot conclude that contextual features do not help, since they are only helpful

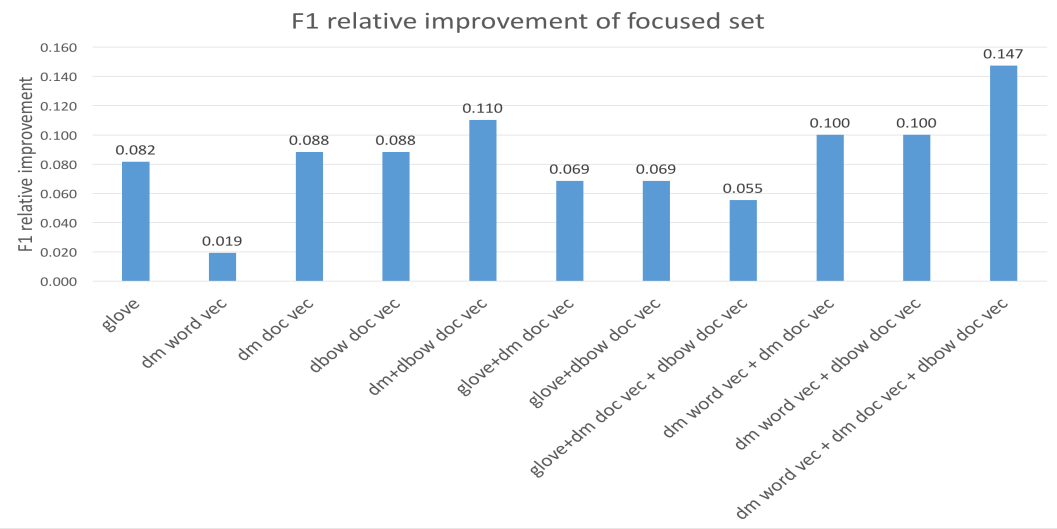


Figure 2: F1 relative improvement on focused set

	baseline	glove	dm word vec	dm doc vec	dbow doc vec	dm doc vec+dbow doc vec
precision	0.239	0.286	0.254	0.267	0.267	0.271
recall	0.484	0.438	0.453	0.500	0.500	0.516
F1	0.320	0.346	0.326	0.348	0.348	0.355

Table 4: Result on focused set when one type of feature is added

when detecting bias associated with ambiguous words. There are only a few ambiguous words in the test set. For non-ambiguous words, the contextual features are not helping but increase the feature dimensionality. It shows that in general cases, the logistic regression model does not learn well when adding the combination of semantic and contextual features.

4.6 Experiment on Focused Set

To better measure the performance of the contextual features in detecting bias associated with ambiguous words, we extracted a focused set of ambiguous words within the test set. We put the word in the focused set if the word is in the training set, labeled as biased at least once, and it is also labeled as not biased at least once. We found words such as "white", "Arabs", "faced", "nationalist" and "black" to be in this focused set. We test our contextual features: dm vector, dbow vector and the combination of the two vectors on the focused set. We also test using the semantic features and the combination of semantic features and contextual features. The result is shown in Tables 4 and 5; the relative improvement of F1 score against the baseline is shown in Figure 2. In the focused set, the maximum F1 score relative improvement of 14.7% is obtained when adding both the dm document vector and dbow document vector combined with dm word vectors.

In the focused set, the advantage of the GloVe feature is not as obvious as in the full test set. Our result shows contextual features (dm document vector + dbow document vector) do help in detecting bias associated with ambiguous words. The model's performance reaches a maximum when the dm document vector and dbow document vector are combined with dm word vector. GloVe features alone behave consistently well in general cases. The result shows the linguistic model behaves better in detecting bias associated with ambiguous words when the contextual information in which the word resides is given. But when we combine GloVe features and contextual features together, the performance gets worse. The performance of the model when GloVe features are combined with contextual features is consistent in both test set and focused set. The result suggests that in bias detection for reference works, we should train two linguistic models: one with added semantic features from either GloVe or the dm model to determine non-ambiguous words' bias detection; one with adding semantic and contextual features learned from dm and dbow models to determine bias associated with ambiguous words. Example 5a was found in the focused set, where it was not predicted correctly by baseline but predicted correctly after dm document vector and dbow document vector are added to train the logistic regression model:

5. (a) According to eyewitnesses, when one of the occupants went to alert the **Israelis** that people were inside, **Israelis** began to shoot at the house.

   (b) According to eyewitnesses, when one of the occupants went to alert the **Israeli soldiers** that people were inside, the **soldiers** began to shoot at the house.

The example was extracted from the Wikipedia article "Zeitoun incident". After we learn the document vector representation of the article "Zeitoun incident" and add it as context when training the linguistic model, the ambiguous word "Israelis" is now recognized as a biased word.

	baseline	GloVe + dm doc vec	GloVe + dbow doc vec	GloVe + dm doc vec + dbow doc vec	dm word vec + dm doc vec	dm word vec + dbow doc vec	dm word vec + dm doc vec + dbow doc vec
precision	0.239	0.280	0.280	0.275	0.271	0.271	0.285
recall	0.484	0.438	0.438	0.438	0.500	0.500	0.516
F1 score	0.320	0.342	0.342	0.337	0.352	0.352	0.367

Table 5: Result on focused set when the combination of two types of features are added

5 Future Work

In this work, we consider vector space representations of text in the bias detection task. Traditional bias detection is usually conducted through manually crafted features as input in a machine learning algorithm such as SVM or logistic regression. After words have been successfully represented as vectors via word analogy, these vectors could be understood by complex language models such as deep neural networks. Future work can consider a deep learning solution for the bias detection task. The solution will be in two phases. Without manually crafted features, in the first phase text in which the target word resides will be input in the neural network model to train vector representations; next the vector representations will be treated as features to train a classifier for bias detection task.

6 Conclusion

In this work, we have noted some drawbacks of using precompiled word lists to detect bias. We use recent research progress in vector space representations of words and documents as semantic features and contextual features to train a logistic regression model for the bias detection task. Our experiment shows that semantic features learned from the GloVe algorithm reach a F1 relative improvement of 26.5% against baseline. In the experiment on a focused set of ambiguously labeled words, the linguistic model reaches the highest gain in F1 score when adding the combination of contextual features learned from the dm and dbow models combined with semantic features learned from the dm model. Semantic features learned from the GloVe algorithm behave consistently well in all experiments. The linguistic model behaves better in detecting bias associated with ambiguous words when the context in which the word resides is given.

References

[Bengio *et al.*, 2006] Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer, 2006.

[Gentzkow and Shapiro, 2010] Matthew Gentzkow and Jesse M Shapiro. What drives media slant? Evidence from US daily newspapers. *Econometrica*, 78(1):35–71, 2010.

[Greenstein and Zhu, 2012] Shane Greenstein and Feng Zhu. Collective intelligence and neutral point of view: the case of Wikipedia. NBER Working Paper 18167, National Bureau of Economic Research, June 2012.

[Iyyer *et al.*, 2014] Mohit Iyyer, Peter Enns, Jordan L Boyd-Graber, and Philip Resnik. Political ideology detection using recursive neural networks. In *Proceedings of the Association for Computational Linguistics*, pages 1113–1122, 2014.

[Kahn *et al.*, 2007] Jeffrey H. Kahn, Renee M. Tobin, Audra E. Massey, and Jennifer A. Anderson. Measuring emotional expression with the linguistic inquiry and word count. *The American Journal of Psychology*, pages 263–286, 2007.

[Le and Mikolov, 2014a] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proc. 31st Int'l Conf. on Machine Learning (ICML)*, pages 1188–1196, June 2014.

[Le and Mikolov, 2014b] Quoc V. Le and Tomas Mikolov. Distributed Representations of Sentences and Documents. *ArXiv e-prints*, May 2014.

[Liu *et al.*, 2005] Bing Liu, Minqing Hu, and Junsheng Cheng. Opinion observer: Analyzing and comparing opinions on the web. In *Proc. 14th Int'l Conf. on World Wide Web (WWW)*, pages 342–351, 2005.

[Marneffe *et al.*, 2006] M. Marneffe, B. Maccartney, and C. Manning. Generating typed dependency parses from phrase structure parses. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC-2006)*, Genoa, Italy, May 2006. European Language Resources Association (ELRA). ACL Anthology Identifier: L06-1260.

[Mikolov *et al.*, 2013a] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient Estimation of Word Representations in Vector Space. *ArXiv e-prints*, January 2013.

[Mikolov *et al.*, 2013b] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Inf. Processing Systems (NIPS)*, pages 3111–3119, 2013.

[Noam, 2008] Cohen Noam. Dont like Palin's Wikipedia story? Change it. *The New York Times*, September 2008.

[Pennebaker *et al.*, 2015] James W. Pennebaker, Ryan L. Boyd, Kayla Jordan, and Kate Blackburn. The development and psychometric properties of LIWC2015. *UT Faculty/Researcher Works*, 2015.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[Recasens *et al.*, 2013] Marta Recasens, Cristian Danescu-Niculescu-Mizil, and Dan Jurafsky. Linguistic models for analyzing and detecting biased language. In *ACL (1)*, pages 1650–1659, 2013.

[Saif *et al.*, 2012] Hassan Saif, Yulan He, and Harith Alani. Semantic sentiment analysis of twitter. In *Proc. 11th Int'l Semantic Web Conf. (ISWC)*, pages 508–524. Springer, 2012.

[Yano *et al.*, 2010] Tae Yano, Philip Resnik, and Noah A. Smith. Shedding (a thousand points of) light on biased language. In *Proc. NAACL HLT Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 152–158, 2010.

Argumentative Ranking

Marco Lippi and **Paolo Sarti** and **Paolo Torroni**
DISI - Università degli Studi di Bologna*

Abstract

There are situations where the information we need to retrieve from a set of documents is expressed in the form of arguments. Recent advances in argumentation mining pave the way for a new type of ranking that addresses such situations and can positively reduce the set of documents one needs to access in order to obtain a satisfactory overview of a given topic. We define and implement a proof-of-concept argumentative ranking prototype, to find that the results it provides can significantly differ from, and possibly improve, those returned by an argumentation-agnostic search engine.

1 Introduction

An argument is, broadly speaking, a claim supported by evidence [22]. Argumentation is the reasoning or dialogical process of producing and evaluating such arguments. It is also the name given to the discipline that studies such processes. Arguments are present in everyday life, so much so to suggest that the need to create and use arguments to convince others is the main driver behind the evolution of human reasoning [11]. It was in fact observed that people are better at reasoning when they communicate through an argumentative context, rather than in an abstract setting. Moreover, arguments are used to convince others. Thus, persuasive communications, editorials, political debates, opinionated blogs, etc. are rich in arguments, which show themselves in diverse formats. Capturing such arguments in a way that enables us to reason from them is a cognitive process that we have been training to do well by evolution. Indeed, in many contexts where we need, for example, to form an opinion on a new topic, especially a controversial one, arguments are exactly what we are looking for.

If detecting arguments is an atavistic ability of the human mind, the automatic detection of arguments instead is a relatively new challenge for computer science. In particular, in the past few years we have witnessed great advancements in a new domain, called *argumentation mining*, which addresses the challenging task of automatically extracting structured arguments from unstructured textual corpora [9].

Therefore, while until recently the perspective of retrieving documents based on their argumentative content would have been utopic, the recent availability of argumentation mining methods and tools [1; 20; 8; 15] makes this vision suddenly more concrete.

Possible applications come to mind easily. A search engine that ranked documents based on the amount of claims about a given topic and of evidence related to such claims would be an invaluable companion for news agencies, journalists, communication departments and cabinet staff, and would be useful even to the random browser, since it would positively narrow down the set of documents that one needs to access in order to obtain a satisfactory overview of the topic.

The aim of this short speculative study is thus to introduce the concept of argumentative ranking, propose an initial portfolio of metrics that can be used to implement it, and offer a first, qualitative assessment of the potential of such a ranking by means of a proof-of-concept prototype and a controlled experiment. We show that argumentative ranking does indeed provide results that are quite different from those that are obtained by a "traditional" search engine.

This work is related to the field of *focused retrieval*, that aims to provide users with direct access to relevant information in retrieved documents [14]. Recently, the IBM Haifa Research Group also proposed a method to perform claim-oriented retrieval of Wikipedia pages [16]. Yet, such approach is only a preparatory step for claim detection, by using a set of handcrafted features that are specifically designed to select documents that are more likely to contain claims (e.g., because they contain "controversy"-related terms or are tagged with special Wikipedia annotations that indicate a controversial content). The approach we propose in this paper, instead, directly addresses the ranking problem in document retrieval, by exploiting the information coming from the claims detected by an argumentation mining system. Differently from the IBM approach, that is tailored to Wikipedia articles, our method can in principle be applied to heterogeneous documents, covering any genre and domain. Our case study is conducted on a collection of newspaper articles retrieved from the New York Times website.

*Contact: marco.lippi3@unibo.it, paolo.sarti2@studio.unibo.it, p.torroni@unibo.it.

2 Argumentation Mining

Argumentation (or argument) mining is the automatic extraction of structured arguments from unstructured textual corpora. It has been argued that building systems endowed with argument mining capabilities would pave the way to a variety of innovative applications [9]. That is confirmed by some important investments made in this area by public and private agencies.[1] This makes us believe that maturing argumentation mining technologies will advance even further in the near future.

The architecture of an argumentation mining system is defined by three crucial aspects: the *argument model* it adopts, the set of *corpora* used for training the system, and the *methodology* exploited in addressing all relevant sub-tasks.

The most popular structured argument model in literature is also the simplest possible model, whereby an argument consists of three distinct parts: a set of premises, sometimes also called *evidence*, a conclusion or *claim*, and an inference from the premises to the conclusion [22].

The works that pioneered this field were strongly connected to the available corpora. Historically, the first application domain was law [21; 12], where the idea was to identify arguments in judgments or other legal documents. Some initial datasets were collections of annotated court cases. Other important datasets are the Dundee corpora[2] and the NoDE benchmark [3], which focus on the relations between arguments. Undoubtedly, the largest available dataset to date was produced within the Debater project and is maintained by IBM Research. It consists of 547 Wikipedia articles [1; 15], organized into 58 topics, and it has been annotated with 2,294 claims and 4,690 evidence facts. Other smaller corpora are available on diverse domains such as persuasive essays [19], comments to articles and forum posts [6], and blog threads [2].

The existing argumentation mining methodologies usually implement a pipeline of subsequent stages [9], which takes in input a raw text document, and produces in output a structured document where arguments are highlighted. The first stage extracts sentences that contain an argument component (claim and/or evidence). The second stage detects the boundaries of each component. The final stage predicts the structure of argumentation, i.e., the support/attack relations between arguments or components. Because we are not interested in predicting the whole argument structure, but only in measuring the amount of arguments in a document, the first stage already provides useful output. To this end, claim/evidence detection has been addressed by a variety of tools, including structured kernel machines [17; 8], binary SVM classifiers [20; 5], logistic regression [7; 15], naïve Bayes [2; 20; 12; 5], and recursive neural networks [18].

[1] See for instance the multi-million IBM Debater project, https://www.research.ibm.com/haifa/dept/vst/mlta_data.shtml, and a large ESPRC on argumentation mining at the University of Dundee, http://www.dundee.ac.uk/news/2015/11million-ai-grant-to-mine-arguments-and-analyse-opinion.php

[2] http://www.arg.dundee.ac.uk/aif-corpora/

3 Ranking by Claims

A classifier such as those used in the first stage of the argumentation mining pipeline typically assigns a score to each sentence of a given document. In the case of claim detection, if the score is positive, the sentence is predicted to contain a claim.[3] An argumentative ranking of documents can be obtained by interpreting the sentence-level information produced by the classifier. We defined five indicators measuring the argumentative content of a document D_i:

- $\sigma_1(D_i)$: Number of sentences in D_i containing claims;
- $\sigma_2(D_i)$: Percentage of sentences in D_i containing claims;
- $\sigma_3(D_i)$: Sum of scores of sentences in D_i containing claims;
- $\sigma_4(D_i)$: Average score of sentences in D_i containing claims;
- $\sigma_5(D_i)$: Sum of scores of sentences in D_i containing claims, divided by the total number of sentences in D_i.

Each indicator $\sigma_j(D_i)$ measures a different aspect of the argumentative content of D_i. There is no absolute reason to prefer one indicator over the other. For example, it is difficult to establish a clear preference between a very short document where almost all the sentences are argumentative, and a lengthier document that contains more claims but also several non-argumentative sentences. Similarly, there are reasons for taking into account the magnitude of the scores, which could bring important additional information to the ranking, but one may also decide to ignore that, and consider simple binary information.

For want of a convincing absolute criterion, we decided to combine all these indicators in a single ranking function through a voting process. Combining different scores into a final ranking function is a typical operation in information retrieval systems (e.g., see [4; 13] and references therein). Given a corpus of M documents $\mathcal{D} = \{D_i\}_{i=1}^{M}$ related to a given query, we first computed the five scores described above $\sigma_j(D_i), j \in \{1, \ldots, 5\}$ for each document D_i, thus building five different rankings. Then, we assigned a set of points π_j to each document, based on each individual ranking, following a non-linear mapping: 25 points to the first document, 20 to the second, 16 to the third, 13 to the fourth, then 11, 10, ..., 1 point to the 5th, 6th, ..., 15th document, and 0 points to the others.[4]

The final score $S(D_i)$ of document D_i is thus obtained by summing the points obtained by the document in each of the five rankings induced by the five indicators:

$$S(D_i) = \sum_{j=1}^{5} \pi_j(D_i) \quad (1)$$

[3] In general, this threshold could be tuned so as to improve the recall or the precision of the classifier.

[4] This is the points scoring system adopted in the FIM Motorcycle Grand Prix World Championship.

4 Experiments

Quantitative evaluations of ranking systems are notoriously hard to obtain, because the key utility measure should be "user happiness", which is greatly influenced by the quality of the returned results (difficult to assess by itself), but also by independent factors, such as speed of response, interface design issues, and the size of the index [10]. We thus decided to perform a qualitative analysis of the output. To this end, we set up an experiment aimed to compare our ranking with the results retrieved by a mainstream search engine, such as Google, and identify cases where the argumentative ranking may satisfy the requests of a user.

We randomly selected 30 key phrases from the controversial topics in the IBM corpus. Of these 30 key phrases, 12 consist of a single word (e.g., abortion, austerity, gambling), and 18 of a short phrase (e.g., affirmative action, national service, wind power). We queried the Google search engine[5] with each one of the key phrases in turn, together with the expression `site:www.nytimes.com`, whose effect is to limit the scope of the search to the New York Times website. We saved the top-10 hits of each key phrase. We then implemented a simple crawler[6] in order to collect a larger set of documents from the New York Times website,[7] using the top-10 Google results as seed pages for the crawler. The crawler's policy was to follow a link if at least one of the following two conditions was met: (1) the link URL contained the searched key phrase; (2) the link was contained in a page in which the searched key phrase appeared at least once. Starting from the selected key phrases and seeds, the crawler downloaded 3,197 articles. We further discarded 11 key phrases, for which less than 20 articles could be retrieved. Table 1 provides details on the dataset.

For each article retrieved by our crawler, we run the claim detection system described in [8].

This setup enabled a qualitative comparison between the search results retrieved by a "traditional" search engine, which is mostly based on features induced by the network topology and website reputation, and the argumentation ranking approach, whose distinguishing feature is its ability to highlight argumentative content by analyzing the linguistic and semantic content of a web page.

Space restrictions allow us to comment on a few interesting cases only.[8] Let us first consider the keyword *gambling*. The top-ranked article according to our system is titled "Majority Back Referendum to Add Casinos, Poll Finds," and it does not appear among the top-10 articles retrieved by Google (see Table 2, top). This article is actually highly argumentative, as it provides many pros and cons with respect to the possibility of opening new casinos in the state of New York. In fact, among the claims retrieved by our system, we find both arguments in favor of expanding casino gambling, as in the following sentence:

[5]The experiments were run on November 10–14, 2015.

[6]We used the open source library `crawler4j`.

[7]http://www.nytimes.com/

[8]All the URLs of the downloaded articles and the results of our ranking systems are available at the following website: http://argumentativeranking.altervista.org.

Table 1: Details on the New York Times corpus developed within this work.

Key phrases	Articles	Claims/Sent.	Claims/Artic.
abortion	485	0.062	2.318
affirmative+action	85	0.106	4.553
asylum	223	0.031	1.466
austerity	172	0.054	2.366
blasphemy	22	0.033	1.091
collective+bargaining	60	0.051	2.200
contraception	53	0.097	3.396
endangered+specie	65	0.033	1.508
gambling	73	0.097	5.096
Gaza	690	0.029	1.228
Holocaust	39	0.030	1.359
Keystone+XL	126	0.051	2.048
Myanmar	296	0.037	1.368
national+service	53	0.035	2.132
nuclear+weapon	260	0.038	1.562
sex+education	43	0.052	2.698
video+game	172	0.035	2.384
wind+power	181	0.058	3.558
year+round+school	75	0.037	2.680

> *Seventy-four percent agreed that allowing the development of casinos would create thousands of jobs, and 65 percent agreed that more casinos would generate significant revenue for the state and for local governments.*

and against these new casino openings, such as this one:

> *And 55 percent agreed that developing casinos would only increase societal problems, like crime and compulsive gambling.*

This controversy is summarized by another sentence, explicitly remarking the presence of arguments in the article:

> *The poll found that voters agree with arguments both for and against expanding casino gambling.*

From Table 2 (top) we can also observe that, for this keyword, Argumentative Ranking and Google have only four top-ranked articles in common out of 10. In general, we observe that Google tends to include more news, chronicle and event-related articles, and we know that the number of backlinks plays a major role. If we consider the percentage of sentences containing claims for each article (column $\%_C$), we observe that Google does not necessarily retrieve argumentative content.

As a second example, we consider the phrase *wind+power*. Table 2 (bottom) shows the top-10 documents ranked by our system and by Google. Also in this case, our top-ranked article is not present in Google results. The article is entitled "Salvation gets cheap" and is a 2014 article containing plenty of argumentative sentences that well describe the debate around the topics of renewable energies and pollution. Some of the paragraphs detected by our systems as containing claims are:

> *Even as the report calls for drastic action to limit emissions of greenhouse gases, it asserts that the*

Table 2: Titles and scores of top-10 documents ranked by our system and by Google for the keywords `gambling` (top) and `wind+power` (bottom). For each article we show the percentage of claims $\%_C$ and the overall score S. Items marked N/A were not retrieved by our crawler.

	Argumentative Ranking	$\%_C$	$S(D_i)$	Google Ranking	$\%_C$	$S(D_i)$
1.	Majority Back Referendum to Add Casinos...	0.32	94	Rein In Online Fantasy Sports Gambling	0.42	82
2.	Rein In Online Fantasy Sports Gambling	0.42	82	The Trouble With Fantasy Sports Gambling	N/A	N/A
3.	Nevada Says It Will Treat Daily Fantasy...	0.23	51	17 People in Three States Are Held in...	N/A	N/A
4.	Cash Drops and Keystrokes: The Dark...	0.13	51	The Dark World of Fantasy Sports and...	N/A	N/A
5.	Will Other Leagues Join N.B.A.? Don't Bet...	0.19	45	Cash Drops and Keystrokes: The Dark...	0.13	51
6.	N.F.L.'s Unsteady Stance on a Tricky...	0.19	39	Nevada Says It Will Treat Daily Fantasy...	0.23	51
7.	As Casino Vote Nears, Bishops Warn of...	0.38	37	Daily Fantasy Sports and the Hidden Cost...	0.14	12
8.	Seeking to Ban Online Betting, G.O.P....	0.20	36	The Perfect Predictability of Gambling...	0.07	0
9.	An Ad Blitz for Fantasy Sports Games, but...	0.14	27	Whitney Wortman and William Gambling	N/A	N/A
10.	In Sharp Pivot for N.B.A., Commissioner...	0.25	25	An Ad Blitz for Fantasy Sports Games, but...	0.14	27

	Argumentative Ranking	$\%_C$	$S(D_i)$	Google Ranking	$\%_C$	$S(D_i)$
1.	Salvation Gets Cheap	0.29	62	Wind Power Spreads Through Turbines...	N/A	N/A
2.	State of the Union Address - 2012 Transcript	0.06	50	Europe Looks Offshore for Wind Power	0.19	15
3.	Wind Power Is Poised to Spread to All States	0.46	50	Wind Power Is Poised to Spread to All States	0.46	50
4.	Tesla Ventures Into Solar Power Storage for...	0.15	46	Procter & Gamble to Run Its Factories...	0.10	0
5.	Glut of Coal-Fired Plants Casts Doubts on...	0.16	43	The Falling Cost of Wind Power	0.10	0
6.	Natural Gas: Abundance of Supply and Debate	0.22	41	Solar and Wind Energy Start to Win on...	0.17	36
7.	Texas Is Wired for Wind Power, and More...	0.16	37	Texas Is Wired for Wind Power, and More...	0.16	37
8.	Solar and Wind Energy Start to Win on...	0.17	36	Tax Credit for Wind Power	N/A	N/A
9.	A Price Tag on Carbon as a Climate Rescue...	0.11	36	A Texas Utility Offers a Nighttime Special...	0.12	0
10.	China Wins in Wind Power, by Its Own Rules	0.20	29	HP to Power Texas Data Centers With...	0.00	0

> *economic impact of such drastic action would be surprisingly small.*
>
> *On the left, you sometimes find environmentalists asserting that to save the planet we must give up on the idea of an ever-growing economy; on the right, you often find assertions that any attempt to limit pollution will have devastating impacts on growth.*
>
> *It's even possible that decarbonizing will take place without special encouragement, but we can't and shouldn't count on that.*

In this case, Argumentative Ranking and Google have only 3 top-ranked articles in common. Again, the reported statistics highlight a marked difference between the argumentative content retrieved by the two systems.

We complemented our analysis by studying the outcome of Google queries when we attached keywords such as `debate`, `argument`, and `opinion`. We obtained mixed results: while such keywords brought up the occasional article with argumentative content, we could not observe a significantly consistent improvement.

5 Conclusions

Motivated by recent advances in argumentation mining, we presented a small, speculative study aimed to define and demonstrate the usefulness of argumentative ranking. As a pilot case study we chose a set of paradigmatic, controversial topics from the IBM argumentation mining corpus, and a largely popular newspaper such as the New York Times. We compared the results obtained by our argumentative ranking system and a traditional, argumentation-agnostic search engine. We found that, in several cases, our system produces a high ranking for documents that are rich in argumentative content but are remarkably excluded from the top Google results. We believe that this new type of ranking could enable a new range of innovative applications fit to diverse domains such as journalism and politics but also law, medicine, and market analysis, as well as increase the quality of search for the random browser. Future work will include a quantitative analysis of the performance of our system, following the contributions of the recent area of focused retrieval.

References

[1] E. Aharoni, A. Polnarov, T. Lavee, D. Hershcovich, R. Levy, R. Rinott, D. Gutfreund, and N. Slonim. A benchmark dataset for automatic detection of claims and evidence in the context of controversial topics. In *Proc. 1st Worksh. on Argumentation Mining*, pages 64–68. ACL, 2014.

[2] O. Biran and O. Rambow. Identifying justifications in written dialogs by classifying text as argumentative. *Int. J. Semantic Computing*, 5(4):363–381, 2011.

[3] E. Cabrio and S. Villata. NoDE: A benchmark of natural language arguments. In *Proc. COMMA 2014*, pages 449–450. IOS Press, 2014.

[4] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proc. WWW '01*, pages 613–622. ACM, 2001.

[5] J. Eckle-Kohler, R. Kluge, and I. Gurevych. On the role of discourse markers for discriminating claims and premises in argumentative discourse. In *Proc. EMNLP*, pages 2236–2242. ACL, 2015.

[6] I. Habernal, J. Eckle-Kohler, and I. Gurevych. Argumentation mining on the web from information seeking perspective. In *Proc. Worksh. Front. Conn. Argum. Theory NLP*, CEUR-WS 1341, 2014.

[7] R. Levy, Y. Bilu, D. Hershcovich, E. Aharoni, and N. Slonim. Context dependent claim detection. In J. Hajic and J. Tsujii, editors, *COLING 2014, Dublin, Ireland*, pages 1489–1500. ACL, 2014.

[8] M. Lippi and P. Torroni. Context-independent claim detection for argument mining. In *Proc. 24th IJCAI*, pages 185–191. AAAI Press, 2015.

[9] M. Lippi and P. Torroni. Argumentation mining: State of the art and emerging trends. *ACM Trans. Internet Technol.*, 16(2):10:1–10:25, Mar. 2016.

[10] C. D. Manning, P. Raghavan, and H. Schütze. Evaluation in information retrieval. In *Introduction to Information Retrieval*, chapter 8. CUP, NY, 2008.

[11] H. Mercier and D. Sperber. Why do humans reason? arguments for an argumentative theory. *Behavioral and Brain Sciences*, 34(02):57–74, 2011.

[12] R. Mochales Palau and M.-F. Moens. Argumentation mining. *Artif. Intell. and Law*, 19(1):1–22, 2011.

[13] R. Nuray and F. Can. Automatic ranking of information retrieval systems using data fusion. *Inform. Process. Manag.*, 42(3):595 – 614, 2006.

[14] J. Pehcevski and J. A. Thom. Evaluating focused retrieval tasks. In *SIGIR 2007 Workshop on Focused Retrieval*, 2007.

[15] R. Rinott, L. Dankin, C. A. Perez, M. M. Khapra, E. Aharoni, and N. Slonim. Show me your evidence - an automatic method for context dependent evidence detection. In *Proc.EMNLP*, pages 440–450. ACL, 2015.

[16] H. Roitman, S. Hummel, E. Rabinovich, B. Sznajder, N. Slonim, and E. Aharoni. On the retrieval of wikipedia articles containing claims on controversial topics. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 991–996. International World Wide Web Conferences Steering Committee, 2016.

[17] N. Rooney, H. Wang, and F. Browne. Applying kernel methods to argumentation mining. In *Proc. 25th FLAIRS*. AAAI Press, 2012.

[18] C. Sardianos, I. M. Katakis, G. Petasis, and V. Karkaletsis. Argument extraction from news. In *Proc. 2nd Worksh. on Argumentation Mining*, pages 56–66. ACL, 2015.

[19] C. Stab and I. Gurevych. Annotating argument components and relations in persuasive essays. In J. Hajic and J. Tsujii, editors, *COLING 2014, Dublin, Ireland*, pages 1501–1510. ACL, 2014.

[20] C. Stab and I. Gurevych. Identifying argumentative discourse structures in persuasive essays. In *Proc. EMNLP*, pages 46–56. ACL, 2014.

[21] S. Teufel. Argumentative zoning. *PhD Thesis, University of Edinburgh*, 1999.

[22] D. Walton. Argumentation theory: A very short introduction. In *Argumentation in Artificial Intelligence*, pages 1–22. Springer US, 2009.

Extracting Predictions and their Scopes from News Articles

Navya Yarrabelly
IIIT-Hyderabad
yarrabelly.navya@research.iiit.ac.in

Kamalakar Karlapalem
IIIT-Hyderabad/IIT-Gandhinagar
kamal@acm.org

Yashaswi Pochampally
IIIT-Hyderabad
p.yashaswi@research.iiit.ac.in

Abstract

We estimate that nearly one third of news articles contain references to future. Distinguishing such predictive statements from factual statements in news articles is important for most applications such as fact checking, opinion mining, future trend analysis, etc. In this paper, we approach a problem of automatically extracting future related information precisely by solving two sub problems. The first sub-problem is labeling a sentence as predictive or factual and the second one is resolving scope of the prediction in a sentence. We formulate a solution to the two sub-problems as a supervised classification task, where we extract all the clauses of a given sentence and classify each of the clauses as predictive or not. We then disambiguate the clause labels to give a label to the sentence. For this we use syntactic structure of the sentence coupled with dependencies within a clause along with dependencies between the clauses. Our solution also deals with annotating prediction statements with condition and factual base (if any), i.e the base on which the prediction is made and the occurrence of the condition on which the validity of a prediction depends.

1 Introduction

Any statement made in reference to future is a prediction[1]. News articles often talk about stock market predictions, economic growth projections or predictions about a future event. These also include future statements referring to global financial crisis, politics, globalization, climate changes, and technologies. Such future trends could have potential impact on our lives, business, etc. Given the attention of readers in analyzing such predictions, isolating and presenting predictions from other news stories improves user engagement with the news site.

Modern style of writing generally has complex compound sentences where predictive statements are generally bloated with their factual bases. In our context, we define factual base for a prediction as knowledge, facts, science, experiments etc, based on which the prediction is being made. Given a prediction, retrieving any predictions or facts related to it has important applications. To formulate a query for this retrieval problem, we need to concisely identify the scope of a predictive part in a sentence for uniquely identifying the terms of a prediction. Isolating the predictive part and its base improves the efficiency of such retrieval model. Given the interestingness of such future related information from both readers and authors perspective, it gives us a motive to automatically extract such predictions concisely with high accuracy from a news corpus.

A sentence with a predictive clause can be predictive or factual depending on its association with other clauses in the sentence. For example consider three excerpts below, extracted from BBC [2] news stories and Times Now[3] news stories.

I Government promised to extend the maternity leave by 2 months by the end of 2005.

II Though the government promised to extend the maternity leave, it could not keep its promise.

III Rajnath Singh told he believed that, with key pre-poll pacts now in place around the country, the party and its allies could win 300 seats of the 543 being contested.

Statement I is a prediction while statement II is a fact, though both the statements have a common predictive clause *"Government promised to extend the maternity leave"*. Statement III is a prediction with predictive clause *"the party and its allies could win 300 seats of the 543 being contested"* and factual base for the prediction *"with key pre-poll pacts now in place around the country"*. From these examples, it is clear that the problem of prediction extraction cannot be solved just by extracting the linguistic patterns or keywords which are predictive. But there is an imperative need to address a solution with a new NLP perspective for processing of a sentence as its constituent clauses and analyze the dependencies between the clauses, which other methods lack in their approach.

Our method includes NLP-processing steps of clause extraction from a sentence and extracting dependencies between

[1] http://www.merriamwebster.com/dictionary/prediction

[2] http://www.bbc.com/news

[3] http://www.timesnow.tv/

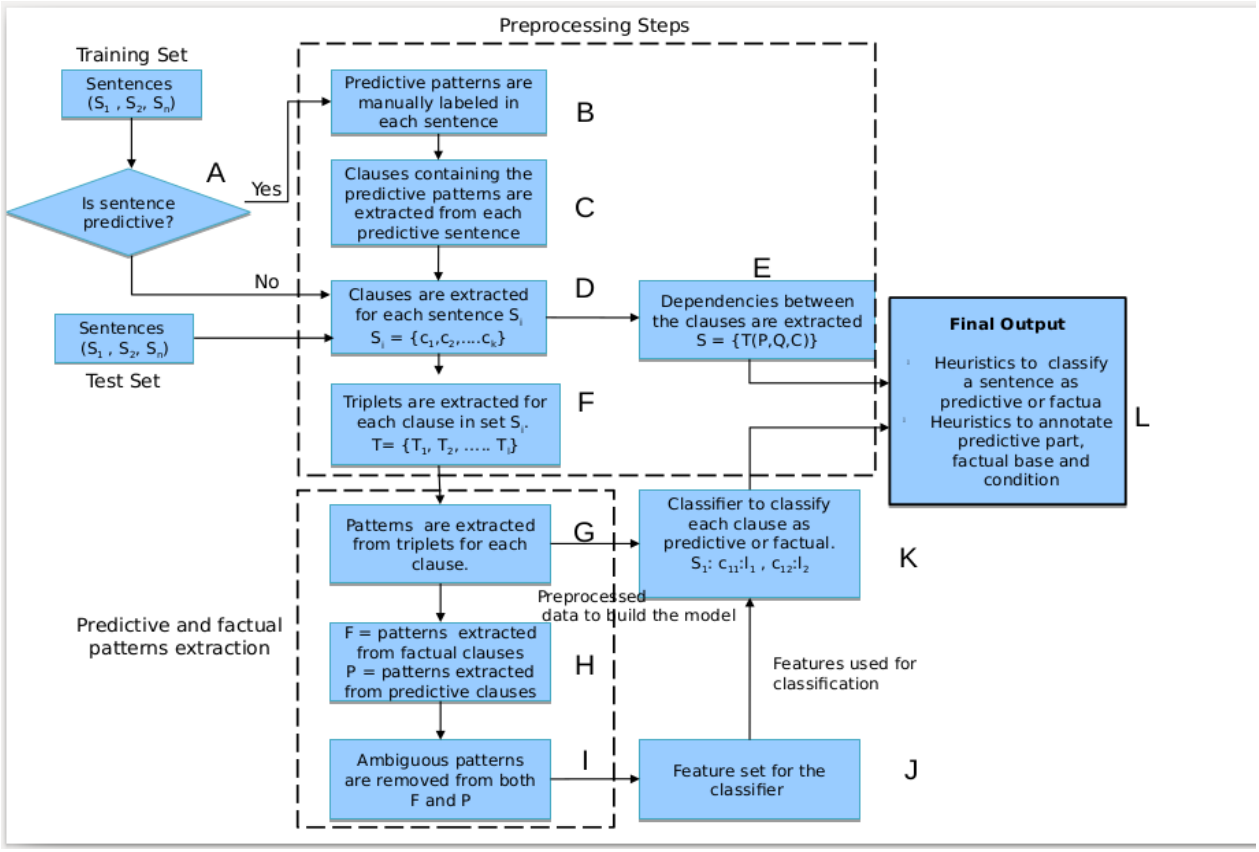


Figure 1: Framework for automatically extracting predictive sentences and resolving the scope of prediction in a sentence.

the clauses from the constituency parse tree. From the dependency tree, open relation triplets are extracted for each clause. From these (subject, predicate , object) triplets, we then extract linguistic patterns which uniquely represent predictive statements and train a classifier with these features. Each sentence is classified as predictive or not from its constituent clause labels, following a set of heuristics.

2 Related Work

Although linguistically expressed references to the future has been studied by a number of researchers, it has only gained interest from a natural language processing perspective in the recent years [Nakajima *et al.*, 2014; 2015].

[Jatowt *et al.*, 2009; 2010; Kawai *et al.*, 2010] extracts and retrieves time-referenced predictions from a given arbitrary query. [Kanhabua *et al.*, 2011] retrieves and ranks predictions that are relevant to a news article. However their prediction models are limited to only sentences referring to future event and temporal expressions and none of them used more sophisticated expressions considering the semantics and sentence structure. These methods suffer from low recall as we estimate from our results (table 1) that only 35% of the predictions are time referenced.

[Nakajima *et al.*, 2014; 2015] generates a list of morpho-semantic patterns that appear uniquely in only future related information. They performed semantic role labeling on sentences to extract features based on ngrams and trained a classifier to classify a sentence as future referenced or not. However, their methods suffered from low precision as it considered a whole sentence as prediction for their classifier to extract future-specific patterns. As mentioned from above examples, a sentence can contain both predictive and factual parts. Hence factual and predictive parts should be separated and processed individually to extract future specific patterns more accurately.

[Özgür and Radev, 2009] detects speculations and their scopes in Scientific Text. It classifies the potential keywords as real speculation keywords or not by using a diverse set of linguistic features that represent the contexts of the keywords and using the syntactic structures of the sentences to determine their scope. We estimate from our results(table 4) and shown from above examples, that in the case of news articles, it is syntactic patterns of a sentence coupled with its structure that makes it predictive more than the presence of keywords.

Unlike previous studies which treat the problem of extracting predictions as sentences which contain expressions or pat-

terns referring to future, we extract features by considering only the predictive parts of a statement while training the model. We also deal with a more complicated problem of identifying the scope of a prediction in complex-compound sentences, which other methods lacked in their approach. Our predictive model involves NLP based approach to classify a sentence by taking into account the syntactic structure of a sentence coupled with its dependencies and semantic representation. We also extract base on which the prediction is made (if any) and condition on which the prediction is valid, by incorporating core-nlp techniques. Figure 1 shows our framework to classify sentences as predictive or factual. We follow steps in Figure 1 in the order A, B, C, D, E, F, G, H, I, J, K, L to classify a sentence as predictive or not.

3 Dataset Preparation

We randomly selected sentences from a set of news articles and labeled the sentences as predictive or factual and if predictive, predictive pattern of the sentence is identified (Steps A,B in Figure 1).
For example the sentence *"Saulius Mikoliunas* ***could also face*** *action after three fans were arrested for throwing coins on the pitch"* is labeled predictive with *'could also face'* as predictive phrase. From such sentences we collected 280 predictive clauses which serve as positive instances and 280 factual clauses to serve as negative instances for the classifier.

Instead of labeling a whole sentence as predictive or not, we have labeled only the phrases which are predictive. The predictive part of a sentence is taken as the clause which contains the predictive phrase identified. This helps us to remove the linguistic patterns which actually belong to the fact part of a predictive sentence but are taken as predictive patterns.

4 Extracting Predictive Clauses

We pre-process the labeled data and train a classifier which labels a given clause as predictive or not.

4.1 Pre-processing the Labeled Data

For all the pre-processing steps we have used Stanford CoreNLP module. [4]

Step C - Scope of the Predictive Phrase

The clause containing the predictive pattern (which is extracted in Step B) is extracted. Let T=$\{t_1, t_2 ... t_n\}$ be the set of nodes in the parse tree, corresponding to the words in predictive phrase . In the parse tree we find a clausal node say t_i, which has both NP and VP subtrees and is the deepest common ancestor to all the nodes in T. This node is taken as the root node and the clause formed by this clausal node is taken as predictive part of the predictive sentence.

Step D - Clause Extraction

Each clause is generally comprised of multiple independent clauses connected by either a coordinating conjunction (denoted by CC node in parse tree) or comma(',') and dependent clauses connected by subordinating conjunction (denoted by SBAR node in parse tree).

[4] http://stanfordnlp.github.io/CoreNLP/

The parse tree is split into two trees at the conjunction node in a top down manner and each tree at the end of this step is taken as a clause extracted from the sentence. Anaphora resolution is performed for each of the clauses. Clauses extracted from predictive part (extracted in Step C) are taken as predictive clauses and other clauses are labeled as factual clauses.

Step E - Clausal Dependencies

Each dependent clause is associated with an independent clause, which it modifies or serves as a component of it. Conjunctions between the clauses signify the relationships between their ideas.

We classify conjunctions to 3 classes based on how it signifies a relationship between the clauses. Say conjunction d connects clauses P and Q, d is classified as follows.

- Opposite : If Q contradicts the idea of P.
  Example: "**Though** the government planned to set up a new high court, it did not keep its promise."
- Reason : If Q acts as a purpose or reason to P.
  Example: "**As** the pre-poll pacts are now in place around the country, BJP could win 300 seats."
- Condition : If P is conditional on Q.
  ''If Modi also contests from Varanasi constituency, Kejriwal may have less chances to win the LS seat. "

Appendix A shows the classification of conjunctions into these 3 classes[5].

We extract a set of clausal dependency relations in a sentence as follows. *T(P,Q,C)* denotes a relation of type T between clause P and clause Q, connected by conjunction of class C.

- If clause Q is dependent on P, then C is the class of the subordinating conjunction between the clauses and type T = dependent.
- If Q and P are independent clauses, then C is the class of the coordinating conjunction between the clauses and type T = independent.

We exploit the clausal dependencies to classify a sentence as predictive or not from the labels assigned to its constituent clauses.

Step F - Triplets Extraction

Each of the clauses extracted above (in Step D) is semantically represented as (subject, predicate, object) triplets using dependency parse tree of the clause. A clause is represented by more than one relation triplets if one of the verbs or nouns in the clause introduces a clausal complement (identified by a dependency governed by one of the relations ccomp, xcomp, advcl in the Stanford dependency tree).
For example triplets extracted from the clause *"Sue promised George to respond to his offer"* are
t_1= (Sue, **promised**, George) and
t_2=(Sue, promised to **respond** to, his offer)

[5] https://en.wikipedia.org/wiki/Adverbial_clause

Step G - Annotating the Predictive Clauses
Each predictive clause is annotated with event expressions and future temporal references using Tarsql toolkit[6] and Stanford SUTime [7].

4.2 Step H - Predictive and Factual Patterns Extraction

With the intuition that the predictive nature of a sentence is defined by the linguistic patterns contained by its predicates, we extracted features from the predicates in triplets to identify the patterns uniquely referring to future. In each triplet the subject and object parts are simply replaced by "subject" and "object". From the processed labeled data, following features are extracted for all the triplets in each clause.

- POS tags : Set of n-grams of POS tags in triplets.
- Word co-occurrences : Set of n-grams of words in the triplets.
- Presence of explicit or implicit future temporal references in the clause.
- Presence of future event references in the clause.

Example-1 : The sentence *"Google has plans to set up a new office in Hyderabad"* has only one clause represented as triplet (Google, has plans to set up , a new office in Hyderabad). To extract patterns, we further represent a triplet as (subject/subject, has/VBZ plans/VBZ to/TO set/VB up/RP, object/object) and ngrams extracted from this triplet are marked as predictive patterns .
Example-2 : *"Google's plans to set up a new office in Hyderabad are blown away"* is a factual clause and has triplets (Google's plans, are blown away,) and (Google's plans, to set up, a new office in Hyderabad). Ngrams from the triplets (subject/subject, are/VBP blown/VBN away/RP, object/object) and (subject/subject, to/TO set/VB up/RP, object/object) are marked as factual patterns.

4.3 Step K - Classification of Clauses

Let P be the set of features extracted from predictive clauses and F be the set of features extracted from factual clauses (in the Step G). Patterns which occurred with frequency less than a threshold of 5 are removed from both the sets (Steps I,J). Table 4 shows sample patterns for both the classes. Each clause is represented as a feature vector, with features from both P and F. Taking the pre-processed labeled data (from Step G) as training set, we trained a classifier to label a clause as predictive or factual.

5 Extracting Predictive Sentences

5.1 Clause Labels Disambiguation

Each clause in a sentence is labeled as predictive or factual by the classifier (in Step K). A sentence may get both predictive and factual labels from its constituent clauses. From the constituent clause labels of a sentence, we follow a set of heuristics to label the sentence. Let S be the set of clausal dependency relations extracted (in Step E). Let $T(P,Q,C)$ from S be a dependency relation between the clauses P and Q connected by a conjunction of class C. L_p and L_q be class labels for the clauses P and Q respectively.

1. If P and Q are independent clauses
   (a) if L_p is predictive and L_q is factual and class C is Opposite, then the sentence is labeled factual, as Q is an independent clause and is contrasting the predictive nature of P.
   Example : *"The government planned to set up a new high court **but** it did not keep its promise"* is factual.
   (b) In all other cases, the sentence is labeled predictive as P and Q are independent clauses.
   Example : *"The government is planning to set up a new high court **and** has acquired the required land for it"* is predictive.
2. If Q is dependent on the clause P
   (a) if L_p is predictive and L_q is factual, then the sentence is labeled predictive. As Q is dependent on P, it can only serve as a component of P but cannot contrast P (an independent clause).
   Example: *"**Despite** Barkley 's three missed penalties in the 18-17 defeat against France , France is expected to retain Barkley at inside centre"* is predictive.
   (b) if L_p is factual and L_q is predictive then the sentence is labeled predictive only if the class C is Reason.
   Example: *"The condition is **so** worse **that** the government will start to impose a Swachh Bharat cess of 0.5%"* is predictive.

5.2 Predictive part, Condition and Base for a Prediction

If a sentence is labeled as predictive and has a clausal dependency relation $T(I,D,C)$, which is marked predictive. L_p and L_q be the class labels for the clauses P and Q respectively, then we extract the presence of factual base or condition for the prediction as follows.

1. If L_q is factual and L_p is predictive and class C is Reason, then the predictive part is P and the factual base for the prediction is Q.
2. If L_p is predictive and L_q is predictive class C is Reason, then the predictive part is P and the condition for the prediction is Q.

Example : Prediction *"Martina Hingis has admitted that Martina Hingis may consider a competitive return to tennis if appearance in Thailand later this month goes well"* has predictive part *"Martina Hingis may consider a competitive return to tennis"*, has condition *"if an appearance in Thailand later this month goes well"* and base *"Martina Hingis has admitted"*.

6 Experiments and Results

In this section we discuss the experiments performed and the results achieved to verify whether the predictive statement extraction method is effective. Figure 1 shows our framework

[6] http://www.timeml.org/index.html
[7] http://nlp.stanford.edu/software/sutime.shtml

to extract predictive sentences and resolving scope of the prediction in a sentence.

6.1 Dataset

We randomly selected sentences from a set of news articles and labeled the sentences as predictive or factual. We created two datasets Set560 from politics domain and Set200 from sports domain. Set200 has 100 predictive clauses from 70 predictive sentences and 100 factual clauses from 58 factual sentences. Set560 has 280 predictive clauses from 180 predictive sentences and 280 factual clauses from 101 factual sentences.

6.2 Classification Results

We first classify each clause of a sentence as predictive or factual. From the labels of clauses in a sentence, we follow a set of heuristics as discussed in section 5.1 to label the sentence. To classify whether a clause is predictive or factual, we built linear SVM models using 10 fold cross validation on Set560. We used various combinations of the features introduced in Section 4.2. Table 1 summarizes the results obtained on classification of clauses for the dataset Set560 for various feature sets. FTR refers to both implicit and explicit future temporal references. Low fscore for FTR features in table 1 marks the need to extract linguistically expressed future referencing patterns to extract predictive statements. Table 2 summarizes the results for classification of sentences as predictive or not on Set560, taking input as clause labels(from Step K) and clausal dependencies(extracted from Step E).

Feature Set	Precision	Recall	Fscore
Bigrams, FTR	**0.944**	**0.902**	**0.922**
Trigrams, FTR	0.941	0.888	0.913
Bigrams, Trigrams, FTR	0.927	0.888	0.907
Bigrams, Trigrams	0.939	0.824	0.877
FTR	0.789	0.353	0.487

Table 1: Accuracy measures for predictive clauses classification

Dataset	Precision	Recall	Fscore
Set 200	0.95	0.86	0.903
Set 560	0.97	0.92	0.942

Table 2: Accuracy measures for predictive sentences classification

As Set200 is extracted from Politics domain and the features from the model is trained on Sports domain(Set 560), a high fscore for this dataset shows that our approach to extract predictions works efficiently for any domain and is not because of over-fitting. We also tried different classifiers like Bayesian Logistic Regression classifier. Both the classifiers gave roughly the same results (**fscore** around **0.85-0.92**). Recall is slightly low compared to precision and can be improved by incorporating more semantics based linguistic features to overcome the vocabulary constraints in the classification model.

We also tried different representations of the predictive part of a predictive sentence and extracted the features. Table 3 summaries the classifications results for various cases. Sentences in table 3 refers to the case where a whole sentence is taken as a prediction instead of taking only the predictive clause of the prediction. Patterns are extracted from whole sentence instead of extracting from triplets of predictive clauses. Triplets in table 3 refers to the case where triplets of a sentence are given as instances to the classifier. Low fscores in these cases imply that, extracting clauses to precisely extract the predictive part of a sentence is efficient. From this, it is clear that pre-processing the predictions to extracts predictive clauses and extracting linguistic patterns from the triplets increases efficiency of the prediction extraction model.

Representation	Precision	Recall	Fscore
Clauses	**0.944**	**0.902**	**0.922**
Sentences	0.912	0.54	0.678
Triplets	0.904	0.647	0.754

Table 3: Accuracy measures for different representations of predictive part of a prediction.

6.3 Extraction of Future Reference Patterns

Apart from the automatic classification of clauses, we are also interested in the actual patterns that influenced those results. We extracted the most frequent unique future-reference patterns and non-future-reference patterns from the experiment based on set560. We obtained 156 patterns for the former and 112 patterns for the latter, after removing ambiguous patterns and less frequent patterns. Example patterns extracted, which are common to both Set560 and Set200 datasets are shown in Table 4.

Count	Predictive Pattern	Count	Factual Pattern
266	MD VB object	204	subject VBD object
172	subject MD VB	56	subject VBZ VBN
75	subject VBZ TO	38	subject VBD VBN
45	VBP VBG TO VB	24	subject VBD IN
32	subject MD VBN	21	MD VB VBN

Table 4: Examples of linguistic patterns extracted

6.4 Results for resolving the scope of a prediction

We also annotate a prediction with the predictive part, factual base for the prediction on which it is made and the condition for the validity of the prediction. Table 5 summarizes the results on Set560.

Feature	Number of predictions annotated with the feature	Accuracy
Resolving predictive scope	116/181	0.94
Extracting factual base	55/181	0.59
Extracting condition on which prediction depends	24/181	0.74

Table 5: Results for annotating a prediction with predictive scope, factual base, Conditional prediction

7 Conclusion and Future Work

We presented an approach to classify a sentence as predictive or factual by extracting clauses from a sentence and exploiting the dependency structure of the clauses. We learnt linguistic patterns which uniquely refer to future. We tested our method on two datasets of different sizes. We found out that the method performs well for both sets (Fscore around 0.91-0.93). Our method also deals with resolving the scope of a prediction in a sentence and also extracting the factual base on which the prediction is made.

In future, we plan to introduce semantics of the words in our prediction extraction model. We also plan to validate the truthfulness of these predictions to give credibility scores for authors of the predictions. To approach this problem, we validate the predictive part of the prediction, by using the base as a context to retrieve the facts related to that prediction.

A Classification of conjunctions

	Opposite	Reason	Condition
Subordinate Conjunctions	even if, whereas, although, though while,even though	because, since, as , so that, in that, in order that,despite	if, once, unless, after, before, as soon as, as long as, since, until, while, when, whenever
coordinate Conjunctions	but, yet	for, so	-

Table 6: Classifying conjunctions to signify its relationship between clauses

References

[Jatowt *et al.*, 2009] Adam Jatowt, Kensuke Kanazawa, Satoshi Oyama, and Katsumi Tanaka. Supporting analysis of future-related information in news archives and the web. In *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, pages 115–124. ACM, 2009.

[Jatowt *et al.*, 2010] Adam Jatowt, Hideki Kawai, Kensuke Kanazawa, Katsumi Tanaka, Kazuo Kunieda, and Keiji Yamada. Analyzing collective view of future, time-referenced events on the web. In *Proceedings of the 19th international conference on World wide web*, pages 1123–1124. ACM, 2010.

[Kanhabua *et al.*, 2011] Nattiya Kanhabua, Roi Blanco, and Michael Matthews. Ranking related news predictions. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 755–764. ACM, 2011.

[Kawai *et al.*, 2010] Hideki Kawai, Adam Jatowt, Katsumi Tanaka, Kazuo Kunieda, and Keiji Yamada. Chronoseeker: Search engine for future and past events. In *Proceedings of the 4th International Conference on Uniquitous Information Management and Communication*, page 25. ACM, 2010.

[Nakajima *et al.*, 2014] Yoko Nakajima, Michal Ptaszynski, Hirotoshi Honma, and Fumito Masui. Investigation of future reference expressions in trend information. In *Proceedings of the 2014 AAAI Spring Symposium Series, Big data becomes personal: knowledge into meaning–For better health, wellness and well-being*, pages 31–38, 2014.

[Nakajima *et al.*, 2015] Yoko Nakajima, Fumito Masui, Hiroshi Yamada, Michal Ptaszynski, and Hirotoshi Honma. Automatic extraction of references to future events from news articles using semantic and morphological information. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 4385–4386. AAAI Press, 2015.

[Özgür and Radev, 2009] Arzucan Özgür and Dragomir R Radev. Detecting speculations and their scopes in scientific text. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1398–1407. Association for Computational Linguistics, 2009.

Annotating Satire in Italian Political Commentaries with Appraisal Theory

Michele Stingo, Rodolfo Delmonte
Department of Language Studies & Department of Computer Science
Ca' Bembo 1075 – Ca' Foscari University – 30123 Venezia (Italy)
stingomichele@gmail.com delmont@unive.it

Abstract

This paper presents annotation work to manually classify satire/sarcasm in long commentaries on Italian politics. It is based on Appraisal Theory and uses some 30K word texts. The underlying hypothesis is that using this framework it is possible to pinpoint precisely the deep semantic contents of evaluative judgements and appreciations making up an ironic comment. We performed a high level annotation using major categories, and then refined the classification starting from the lexica derived from the xml annotated files. In this way we succeeded in differentiating texts by the two authors we chose, one of which is characterized by a sharp cutting ironic/almost sarcastic style.

1 Introduction

We present work carried out on journalistic political commentaries in two Italian newspapers, by two well-known Italian journalists, Maria Novella Oppo, a woman, and Michele Serra, a man[1]. Political commentaries published on a daily basis consists of short texts not exceeding 400 words each. Sixty-four texts come from Michele Serra's series titled "L'Amaca", published daily on the newspaper "La Repubblica" between 2013 and 2014; usually the targeted subjects are politicians, bad social habits and in general every trendy current event. Forty-nine texts come from Maria Novella Oppo's series titled "Fronte del video", published daily on the newspaper "L'Unità" in a previous span of time, say from 2011 to 2012; the targeted subjects are usually politicians and televised political talk shows.

The two journalists have been chosen for specific reasons: Oppo is a master in highly cutting and caustic writing, Serra is less so. Both are humorous, however, Oppo is more witty in building the overall logical structure of the underlying satiric network of connections. Oppo borders sarcasm, Serra never does so. Oppo's texts are slightly longer than Serra's.

Italy is not included in the upper part of the list of countries where the freedom of information is very highly regarded. Because of her trenchant and stinging style, Oppo has been publicly attacked by some of her favourite targets, politicians including Berlusconi and Beppe Grillo - two well-known political leaders, who reacted bitterly to her commentaries. In particular, the second one included her in a black list of journalists criticizing his movement called Movimento5Stelle. As a result, she has been attacked – and heavily offended - by Grillo who claimed that since she has always been working for the same newspaper for all her career, she will be out of her job in case the Parliament approves the law that precludes newspapers from receiving public financial support. And she will be obliged to find a new job[2]. Criticisms to the attack has come from many sources[3]. Berlusconi wanted to sue the newspaper l'Unità where Oppo published her commentaries, but the action didn't result in a real lawsuit for defamation or libel action, and ended up being regarded a case of wrongful prosecution.

In order to focus on the specific features connotating political satire, manual annotation has been carried out on 112 texts using a reduced (and modified with new criteria, where needed) version of the Appraisal Framework [Martin & White, 2005].[4] Below and in the next two sections we will delve into a precise description of both the framework and the specific criteria devised for our annotation. Then in a final section we show results and comparisons derived from our annotation work.

2. Satire and the Appraisal Framework

The decision of adopting Appraisal Theory (hence APTH) is based on the fact that previous approaches to detect irony - a word we will use to refer to satire/sarcasm - in texts have failed to explain the phenomenon. Computational research on the topic has been based on the use of shallow features, as in [Carvalho et al., 2009; Burfoot & Baldwin, 2009; Davidov et al., 2010; Reyes & Rosso, 2011; Owais et al., 2015]. This has been done in order to train statistical model with the hope that when optimized for a particular task, they would come up with a reasonably acceptable performance. However, they would not explain the reason why a particular Twitter snippet or short Facebook text has been evaluated as containing satiric/sarcastic expressions. Except perhaps for

[1] Permission to republish excerpts from their articles has been granted personally by the authors.

[2] http://www.beppegrillo.it/2013/12/giornalista_del_giorno_maria_novella_oppo_lunita.html

[3] as for instance in http://www.articolo21.org/2013/12/grillo-giu-le-mani-da-maria-novella-oppo/

[4] The annotated corpus was created as part of the master thesis project (inevitably) written by the first author only. As a consequence, no interannotation agreement measurement could have been provided in the present paper.

features based on text exterior appearance, i.e. use of specific emoticons, use of exaggerations, use of unusually long orthographic forms, etc. which however is not applicable to the political satire texts. These texts are long texts, from 200 to 400 words long and do not compare with previous experiments.

The other common approach used to detect irony, in the majority of the cases, is based on polarity detection. So-called Sentiment Analysis is in fact an indiscriminate labeling of texts either on a lexicon basis or on a supervised feature basis [Gianti et al., 2012; Bosco et al., 2015; Hernandez Farias et al., 2015; Özdemir & Bergler, 2015], where in both cases, it is just a binary decision that has to be taken. This is again not explanatory of the phenomenon and will not help in understanding what is it that causes humorous reactions to the reading of an ironic piece of text. It certainly is of no help in deciding which phrases, clauses or just multiwords or simply words, contribute to create the ironic meaning.

By adopting Appraisal analysis, we intended not only to describe but also to compute with some specificity the linguistic regularities which constitute the evaluative styles or keys of political journalistic texts. The theory put forward by [White and Martin, 2005] (hence M&W) makes available an extended number of semantically and pragmatically motivated annotation schemes that can be applied to any text and can be used to draw precise conclusions. In particular, one preliminary hypothesis would be being able to ascertain whether the text under analysis is just a simple report, a report with criticism, a report with criticism and condemnation. This is something that can be established in a totally safe and stable manner by simply counting and comparing the type of categories and subcategories present in the annotations of the text. In the book by M&W there's a neat distinction between three types of voices: 'reporter voice', 'correspondent voice' and 'commentator voice'. Only the commentator voice has the possibility to condemn, criticize and report at the same time, and since we assume that satire, and even more, sarcasm have a strong component made of social moral sanction, we are automatically selecting this as the target of our research hypothesis.

In M&W, the evaluative field called Attitude is organized into three subclasses, Affect, Appreciation and Judgement, and it is just the latter one that contains subcategories that fit our hypothesis. We are referring first of all to Judgement which alone can allow social moral sanction, and to its subdivision into two subfields, Social Esteem and Social Sanction. In particular, whereas Social Esteem extends from Admiration/Admire vs Criticism/Criticise, Social Sanction deals with Praise vs Condemn. As reported in M&W p.52 "… Judgements of esteem have to do with 'normality' (how unusual someone is), 'capacity' (how capable they are) and 'tenacity' (how resolute they are); judgements of sanction have to do with 'veracity' (how truthful someone is) and 'propriety' (how ethical someone is). Social esteem tends to be policed in the oral culture, through chat, gossip, jokes and stories of various kinds – with humour often having a critical role to play... Sharing values in this area is critical to the formation of social networks (family, friends, colleagues, etc.). Social sanction on the other hand is more often codified in writing, as edicts, decrees, rules, regulations and laws about how to behave as surveilled by church and state – with penalties and punishments as levers against those not complying with the code. Sharing values in this area underpins civic duty and religious observances."

The texts we have annotated show the use of any type of judgement, expressed directly by the writer. As M&W (p.170) define it, the "commentator voice" is an evaluative style typically only of commentaries, opinions and editorials. "It is typical of this category in being primarily concerned with assessments of social sanction, but with also making some reference to assessments of social esteem."(ibid.p.170) And further on, we read, "It would seem that within broadsheet journalistic discourse, this function of 'sanctioning' – whether it be via attitudinal assessments or via directives (modals of obligation) – is confined to the one journalistic role, that of commentator. Even though the correspondent voice writer may argue and evaluate, they typically refrain from either mode of 'sanctioning'."(ibid.p.181)

So eventually in our texts we are dealing with the "commentator voice", which may consist of authorial social sanction, plus authorial directives (proposals), in addition to criticism. All the annotations have been done by the first author for his Master thesis and have been counterchecked by second author.

3 Annotating Italian Political Journalistic Texts

For our annotation work we limited ourselves to using one single subsystem. The Attitude subsystem describes the author's feelings as they are conveyed within the text, and it is articulated into three main semantic regions with their relative positive/negative polarity, namely:

- Affect: describes proper feelings and any emotional reaction within the text aimed towards human behaviour/process and phenomena.
- Judgement: considers the ethical evaluation on people and their behaviours.
- Appreciation: represent any aesthetic evaluation of things, both man-made and natural phenomena.

The choice to rule-out the others two subsystem (Engagement and Graduation) and the features of the three sub-categories of the Attitude subsystem, was made mainly to maintain the notational work on a manageable level, and also because we were more interested in a coarse quantitative substantiation of the authors' opinions within the analyzed texts, rather than conducting a fine-grained analysis about their construction or graduation. In other words, we wanted to assess how descriptive a plain

recognition of evaluative sequences is without further detailed information.

Besides, annotation using the Appraisal framework is very hard and highly demanding in terms of decision choice. Differentiating between the three subclasses was done keeping in mind the lexicon and the structures associated to them in the book by M&W. However, we had to translate all the vocabulary into Italian in order to make it easier to use.

3.2 Using the XML format

The annotation work on the texts has been accomplished using the Extensible Markup Language due to its flexibility and because of the possibility to use specifically devised tags. Following there is a snippet of the XML annotation.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<text>
    <p>
        <s>
        Pare che chiamare un taxi a Firenze, ieri, fosse<apprsl
        appreciation="negative">impossibile</apprsl>a causa di
        un<apprsl appreciation="negative">malizioso "guasto"</apprsl>
        dei call-center forse provocato dall'annuncio di nuove licenze:
        cosa che fa<apprsl attitude="negative">inferocire</apprsl>i
        detentori di quelle vecchie.
        </s>
    <!-- other sentences -->
    </p>
    <!-- other paragraphs -->
</text>
```

The tags we used for the annotation include a tag for <text> contains the whole text of the article; <p> to mark paragraphs, and <s> to mark sentences. However, every time the article was published as a unique block of text, we structured the article content, first identifying the sentences and then grouping them in relation to their meaning: when a sentence was strictly related to one or more of the following/previous propositions, they were clustered together within the same paragraph.

Focusing on the annotation of the evaluative sequences instead, every time we found an evaluative word (or sequence of words) within a political satire article, we delimited the item/phrase within the tags <apprsl></apprsl>. Subsequently, following the general indications mentioned above provided by M&W, we assigned one of the three subcategories – affect[5], judgement and appreciation – as attribute of the tag <apprsl>, also providing the positive/negative sentiment orientation as value of the attribute.

3.3 Linguistic Criteria for Annotation

Since the text typology we annotated showed a lot of complex linguistic features, and because no previous work was found on labelling long satiric texts using the Appraisal Framework – nor in the computational linguistics framework, we had to address the annotation task using brand-new criteria specifically designed for the purpose of isolating as many evaluative items/sequences as possible. The criteria, in relation to their most relevant linguistic aspects, are grouped in one of the following set of notational principles, namely lexical, semantic and syntactic set.

Lexical criteria: these notational principles mostly correspond to the indications contained in M&W:

- Whenever an item implicitly or explicitly indicates or presumes an emotive reaction, a mood or a feeling related to the author or to others subjects mentioned by the author, use the tag <apprsl> with the AFFECT attribute and its relative polarity.
- Whenever an item indicates or presumes a judgement on people, groups or actions related implicitly/explicitly to people or groups, use the tag <apprsl> with the JUDGEMENT attribute and its relative polarity.
- Whenever an item indicates or presumes an evaluation on abstract entities, natural phenomena, artificial processes or man-made things, use the tag <apprsl> with the APPRECIATION attribute and its relative polarity.
- The polarity orientation assignment is based on the literal meanings of the evaluative item.
- In case of doubtful polarity orientation, polarity can be assigned by looking at previous or current phrasal context where the evaluative item appears.

Furthermore the phrasal contexts often served not only as clue for the polarity assignment, but they themselves contained evaluative sequences and thus we had to annotate chains of lexical items as single evaluative units. This aspect reflects the discursive nature of long satiric texts. So a number of semantic and syntactic criteria were needed so as to enhance the notational analysis.

Semantic criteria:

- Anytime one or more verb/noun modifiers are found, when they do not represent meaningful evaluation by themselves, they are annotated together with the part of speech that they contribute to modify.
- Any instance of evaluation conveyed by means of a multiword expression, is annotated as a single appraisal unit.
- Any instance of evaluation conveyed by means of rhetorical or figurative language, is annotated as a single appraisal unit. When possible the evaluations are embedded so as to include appraisal units into bigger evaluative unit, in order to fully capture figures of speech such as oxymora, apagoges, rhetorical questions, interjections and the like.

Syntactic Criteria:

- Without exceeding the length of the proposition, it is allowed to annotate phrases as single appraisal unit up until a clause-level, whenever they express opinions or evaluations. Additionally, for those cases where complex phrasal structures were found, we limited ourselves to the annotation of the most evaluative part within the overall sequence, so as to avoid overproduction of long annotation.
- Again, when possible, the clauses have been de-structured

[5] Please notice that we have substituted the name of the **affect** subcategory with the title of the subsystem **attitude**. However, the attribute attitude in our work is equivalent to the subcategory affect.

so that through embedding we were able to capture the evaluation on a clause-level in greater detail.
- It is allowed to annotate evaluative sequences on a clause level even beyond the punctuation marks limits. However, these annotations were very rare.
- In case of dyad/triad of items, whenever they share the same attribute and the same polarity orientation, they are annotated as single evaluative units.
- In case of more than three items in a row that share the same attribute and the same polarity orientation, they were annotated separately.

3.4 Embedded classifications

We created embedded classifications in order to account for the dependency existing between two adjacent phrases in the definition of the literal/nonliteral meaning of the sentence. We counted 220 such embeddings for Serra's texts and 146 for Oppo's texts. Consider a few examples taken from Serra's texts:

<apprsl appreciation="positive">Di scienza si vive</apprsl>, ma<apprsl appreciation="negative">di "allarmi" si muore</apprsl>,<apprsl appreciation="negative">ne ammazza più l'<apprsl attitude="negative">ansia</apprsl>del colesterolo</apprsl>. / One can live of science, one dies from alarms, more get killed by anxiety than by cholesterol.

In this case, the polarity of the embedded annotations is identical as it is for the majority of the cases in Serra's texts. But look at one of the non-identical cases:

Chiunque ci abbia provato, almeno negli ultimi due secoli,<apprsl judgement="positive">ha vinto qualche battaglia<apprsl judgement="negative">ma alla fine ha perduto la guerra</apprsl></apprsl>. / All those who have tried, at least in the last two centuries, have won some battle but at the end have lost the war.

And here below two examples taken from Oppo's texts where we see that the same technique is used:

Intanto, Berlusconi<apprsl judgement="negative"> dilaga in prima persona</apprsl>, <apprsl judgement="negative">sotto forma di una<apprsl appreciation="negative">generale regressione nazionale</apprsl></apprsl>, / In the meantime, Berlusconi floods everywhere in first person under the guise of a general national regression.

where we find in both case the same polarity – negative – but a different category, the first Judgement and the second Appreciation. Now a second example where polarity is also reversed but also category is modified:

E tutto per le<apprsl appreciation="negative">famose cene<apprsl appreciation="positive"> eleganti</apprsl></apprsl>, ragazza alla quale,<apprsl judgement="positive">al massimo, venivano pagati il viaggio e un<apprsl appreciation="negative"> abituccio</apprsl>di circostanza</apprsl>. / And all for the famous elegant dinners, a girl to whom at most the trip was reimbursed and a valueless courtesy dress

A further example that contains a well constructed definition of irony:

Ovvio che lo stile è<apprsl appreciation="positive"> molto diverso</apprsl>: da parte del professore<apprsl judgement="positive">nessuna volgarità</apprsl> e<apprsl judgement="positive">tanto meno barzellette<apprsl appreciation="negative"> sconce</apprsl></apprsl>;<apprsl judgement="positive" >soltanto una ironia<apprsl appreciation="positive">così sottile che <apprsl appreciation="negative">sembra la lama di un coltello<apprsl appreciation="positive">ben affilato</apprsl></apprsl></apprsl></apprsl>. /Obviously, the style is very different: from the side of the professor, no vulgarity and not even dirty puns and jokes, just a subtle irony, so sharp that it seems the edge of a knife well sharpened.

4 Results

The starting hypothesis was that both commentators were characterized by a high number of Judgements and possibily, negative ones. Then we also hypothesized that there should be an important difference between the two corpora, Oppo's being the one with the highest number. These hypotheses have been borne out by the results of the annotation as can be seen in the distribution of categories in the tables presented below. First of all general data about the annotations:

	NoSents	No.Toks	No.Annots
Oppo	**514**	**14350**	**1651**
Serra	**561**	**14641**	**1849**

Table1: Serra's annotations split by polarity

When computing general data for main categories the picture was the one in Fig. 1 below.

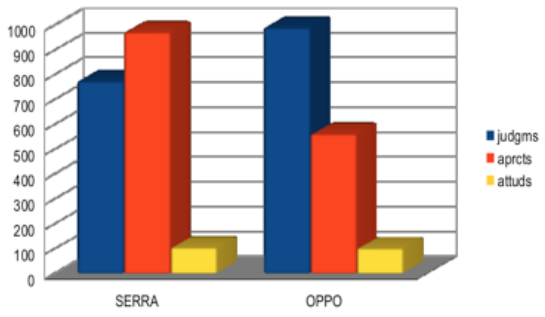


Fig1. : Total Annotations divided by main appraisal classes

There is a clear difference in the use of appraisal evaluative classes in our two authors: Serra seems to prefer Appreciations, Oppo on the contrary favours the use of Judgements. In addition, when we collapsed polarity within the categories we obtained the distribution reported in the Tables 2. and 3. below:

Serra	JudgNegat	JudgPost	ApprNegat	ApprPost
totals	577	216	678	385
mean	9.0	3.4	10.6	6.0
dev.stand.	5.0694	2.6934	4.566	3.6274

Table2: Serra's annotations split by polarity

Oppo	JudgNegat	JudgPost	ApprNegat	ApprPost
totals	824	260	442	188
mean	17.2	5.4	9.2	3.9
dev.stand.	5.289	3.637	3.978	2.727

Table3: Oppo's annotations split by polarity

In Table2. and 3. we report data related to the two main categories collapsed separately by polarity. As can be noted, differences in total occurrencies of Negative Judgements are very high now and Oppo has the highest. Also Positive Judgements shows a majority of cases annotated for Oppo's texts.

On the contrary, with the Appreciation class the difference is in favour of Serra, both for Negative and Positive polarity values. Standard Deviations are higher for Serra's data but this may be due to the disparity of total occurrencies, which in the case of Positive polarity is over the double and in the case of Negative polarity it is about one third higher. Eventually, we can see that Oppo's commentaries are based mainly on Judgement categories and their polarity is for the majority of the cases Negatively marked. Also Appreciation has a strong Negative bias as can be gathered from Table 3. On the contrary, Serra's commentaries are more based on Appreciation and polarity is almost identically biased.

We decided then to recalculate quantitative data using a corrective factor based on total number of tokens and a Normalized Mean Sum. The Normalized Mean Sum is the Sum of all occurrencies divided up by the Sum of the Mean + Standard Deviation. Standard Deviation has now been obtained by multiplying original absolute values with a factor derived from the division of number of annotations in a text by number of tokens. In this way, the absolute values are now comparable between the two set of texts notwithstanding the fact that they are in different number. We report tables related only to Judgements.

	Oppo_JudgN	Oppo_JudgP
SUM	824	260
MEAN	17.1	5.4
DEV.ST	0.681435284	0.395179318
NORMSUM	46.16308581	44.73621646

Table4: Oppo's annotations recalculated on the basis of tokens

	Serra_JudgN	Serra_JudgP
SUM	577	216
MEAN	9.0	3.4
DEV.ST	0.749664666	0.366263499
NORMSUM	59.08682894	57.73450602

Table5: Serra's annotations recalculated on the basis of tokens

As can be seen from the Normalized Mean Sum now it is the case that Oppo's data are much more homogeneously distributed, having a lower Standard Deviation.

4.1 Previous experiments with literal interpretation

The previous analyses of irony was fully literal and was based on the algorithm we used in a number of previous task for sentiment analysis in Italian newspapers in politically related articles. The system uses a number of freely downloadable resources including SentiWordNet, a translated version of Linguistic Enquiry, etc. The results are shown below.

Author	*Voc_Rich*	*Mean_SentLen*	*noToks*	*noSents*	*noRefs*	*Nonfact_prop*
M_Serra	0.1354	0.0391	16820	659	2278	38.1029
MN_Oppo	0.1361	0.034	15673	533	2133	34.2005

Table10: General quantitative data from IT_Getaruns

In this table we show general quantitative measurements where we can see that so-called Vocabulary Richness for the two authors is practically the same. A first interesting result is the ratio of Non-factual propositions compared to factual ones, where we see that the proportion is higher in Serra. In Appraisal Theory terms this amounts to saying that Oppo's level of Engagement is higher than the one of Serra. Number of Referential Expressions used is on the contrary almost identical. The next series of data computed includes polarity values for the two corpora:

Author	*Negat_prop*	*subjectiv*	*p_diathes*	*sp_pos_w*	*sp_neg_w*	*pov*
M_Serra	9.9716	25.2533	1.3782	55.235	44.765	0.8906
MN_Oppo	17.7778	23.794	2.2764	56.6879	43.3121	0.8809

Table11: General quantitative data from IT_Getaruns

The values listed in the table represent ratios computed between couple of attributes. Ratios of negative propositions is computed with respect to positive ones: as can be seen, it is almost the double in Oppo. On the contrary Negative words show the opposite distribution. P_diathesis computes number of passive sentences, which is higher in Oppo, on the contrary Subjective propositions are in the opposite distribution, more in Serra's texts. Negative words have

almost identical distributions and the same applied to positive words. The final attribute POV, evaluates Point of View, if a sentence expresses the Pov of a different character or the Pov of the author: the percentage shows a higher Pov by the author in Serra's commentaries. No clear-cut classification can be gathered from the comparison between annotation data and these ones and in fact we have not even tried correlation measurements of polarities values.

5 Conclusion

As previous scientific literature on the topic suggests – [Taboada & Grieve, 2004; Fletcher & Patrick 2005; Khoo et al., 2012; Read & Carrol, 2012; Hall & Sheyholislami, 2013] – using (a reduced version of) the Appraisal framework proved to be a useful tool for the completion of manual annotation and for further automatic operations. Yet we were not able to represent properly some of the evaluative sequences because of the high level of complexity of the textual structure.

One of the main issue was represented by cases of linguistic cohesion realized through nominal anaphora. Additionally, if we consider that anaphora is not always realized within a sentence, a further level of complexity is added to the representation of evaluative information: the need to take into account discourse or text level anaphora. Future research goals will focus on solving the above issues, further increasing the number of satirical articles included within the corpus while strengthening the manual annotation with the efforts of new annotators and the revision of the existing corpus for a better harmonization with the new texts.

References

{Bosco et al., 2015} C. Bosco, V. Patti, A. Bolioli. Developing Corpora for Sentiment Analysis: The Case of Irony and Senti-TUT (Extended Abstract). In Q. Yang, & M. Wooldridge (Ed.), *Proc. of 24th International Joint Conference on Artificial Intelligence*, IJCAI 2015, 4158 - 4162.

{Burfoot and Baldwin, 2009} Burfoot, C., & Baldwin, T. Automatic satire detection: are you having a laugh? *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, 161--164.

{Carvalho et al., 2009} P. Carvalho, L. Sarmento, M. Silva, E. de Oliveira. Clues for detecting irony in user-generated contents: oh...!! it's so easy;-). *Proceeding of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, ACM, 53-56.

{Davidov et al., 2010} D. Davidov, O. Tsur, and A. Rappoport. Semi-supervised recognition of sarcastic sentences in Twitter and Amazon. *CoNLL '10 Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, 107-116.

{Fletcher and Patrick, 2005} J. Fletcher and J. Patrick. Evaluating the utility of appraisal hierarchies as a method for sentiment classification. *Proceedings of the Australasian Language Technology Workshop*, 134-142.

{Gianti et al., 2012} A. Gianti, C. Bosco, V. Patti, A. Bolioli, L. Di Caro. Annotating Irony in a Novel Italian Corpus for Sentiment Analysis. In L. Devillers, B. Schuller, A. Batliner, P. Rosso, E. Douglas- Cowie, R. Cowie, & C. Pelachaud (Ed.), *Proc. of the 4th International Workshop on Corpora for Research on Emotion Sentiment & Social Signals* (ES3@LREC'12), 1-7.

{Hall and Sheyholislami, 2013} C. Hall, and J. Sheyholislami. Using Appraisal Theory to Understand Rater Values: An Examination of Rater Comments on ESL Test Essays. *The Journal of Writing Assesement*, Volume 6 (1) .

{Hernandez et al., 2015} D. Hernandez Farias, E. Sulis, V. Patti, G. Ruffo, C. Bosco. ValenTo: Sentiment Analysis of Figurative Language Tweets with Irony and Sarcasm. *Proceedings of the 9th International Workshop on Semantic Evaluation* (SemEval 2015), 694-698.

{Khoo et al., 2012} C. Khoo, A. Nourbakhsh, J. Na. Sentiment analysis of online news text: A case study of appraisal theory. *Online Information Review* 36(6).

{Martin and White, 2005} J. Martin and P. R. White. *Language of Evaluation, Appraisal in English.* London & New York: Palgrave Macmillan.

{Owais et al., 2015} Owais, S., Nafis, T., and S. Khanna. An Improved Method for Detection of Satire from User-Generated Content. *International Journal of Computer Science and Information Technologies*, Vol. 6 (3), 2084-2088.

{Özdemir and Bergler, 2015} C. Özdemir and S. Bergler. CLaC-SentiPipe: SemEval2015 Subtasks 10 B,E, and Task 11. *Proceedings of the 9th International Workshop on Semantic Evaluation* (SemEval 2015), 479-485.

{Read and Carrol, 2012} J. Read and J. Carrol. Annotating expressions of Appraisal in English. Language Resources and Evaluation, Volume 46, 421-447.

{Reyes and Rosso, 2011} A. Reyes and P. Rosso. Mining subjective knowledge from customer reviews: a specific case of irony detection. *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis*, 118-124.

{Taboada and Grieve, 2004} M. Taboada and J. Grieve. Analyzing appraisal automatically. In *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect*, 158-161.

An exploratory analysis of news trends on twitter

Konstantinos Bougiatiotis, Anastasia Krithara, George Paliouras and George Giannakopoulos
National Center for Scientific Research "Demokritos", Athens, Greece
{bogas.ko, akrithara, paliourg, ggianna}@iit.demokritos.gr

Abstract

Analyzing social media data can be a rich supplement to the traditional reporting tools of interviews and observation. In this work we proposed a framework for analyzing and exploring twitter streams, by fusing information about the influence of the users, the topics of discussion, the relations and co-occurrences of the named entities. The framework offers expressive visualization tools, enabling users to draw useful conclusions about the data.

1 Introduction

As more people have turned to social-media platforms as a place to gather and share ideas, many journalists have been urged to use these spaces as a place to share their work and indentify important information published. Social media are part of the Big Data paradigm and are characterized by high Velocity, Veracity and Volume ("the 3 Vs"). In Twitter for example, more than 255 million active users publish over 500 million 140-character "tweets" every day[1]. Evidently it has become an important communication medium. More and more people use social media not only to communicate their ideas and thoughts, but also to spread important news. Given the enormous size of information exchange happening every day, it is a rather challenging task for journalists to process these data and filter out the important and relevant information. Analyzing web traffic and social media patterns can be a rich, and increasingly vital, supplement to the traditional reporting tools of interviews and observation. Such data can provide context and clues that also can help better frame and situate stories, as well as furnish new pathways to sources and assess popularity, importance and visibility within the online world.

The topics on Twitter span cross multiple domains from private issues to important public events in the society. Therefore, filtering out the important or relevant to the user information poses the first challenge for automated processing of tweets. For this reason one needs to deploy intelligent methods for focused analysis of all types of content, strongly based on entity and relation extraction techniques, so that information can be clustered around automatically extracted and dynamically evolving entities and relations between them. In addition, another very important issue in social network analysis is the identification of key persons in a social network.

In this work, we propose a framework for analyzing and visualizing the important information from a twitter stream. This framework is based on both natural language processing tools as well as structural analysis in order to identify the important information spread in twitter. It offers a set on expressive visualization tools which can give insights to the user about the analyzed content.

[1] https://about.twitter.com/company

2 Proposed Method

2.1 General Workflow

The overall scheme of the framework described in the current work is depicted in figure 1. The main steps involved in extracting knowledge from semantically-diverse sources and fusing it into meaningful visualizations are the following:

- *Text Analysis*: Using natural language processing techniques, our aim is to extract information about important events unfolding in the tweets. To this end, we deploy a Named Entity Extraction scheme based on the assumption that the main protagonists of such events are Named Entities, like persons or organizations and a Relation Extraction module for identifying the events in the stream of news.
- *Structural Analysis*: The goal of this methodology is to pinpoint the influential users of the network, while taking into account the content of the tweets. To do so, we exploit the structural information of the network while still considering the topic content of the tweets, implementing a hybrid method based on both structure of the network and content of the interactions to rank the influence of the users.
- *Visualization Routines*: Finally, fusing information about the influence of the users, the topics of discussion, the relations and co-occurrences of the Named Entities, we can organize and combine the data into expressive visualizations, enabling us to draw conclusions or get insights regarding the data.

The aforementioned methods are explained in detail in the following sections.

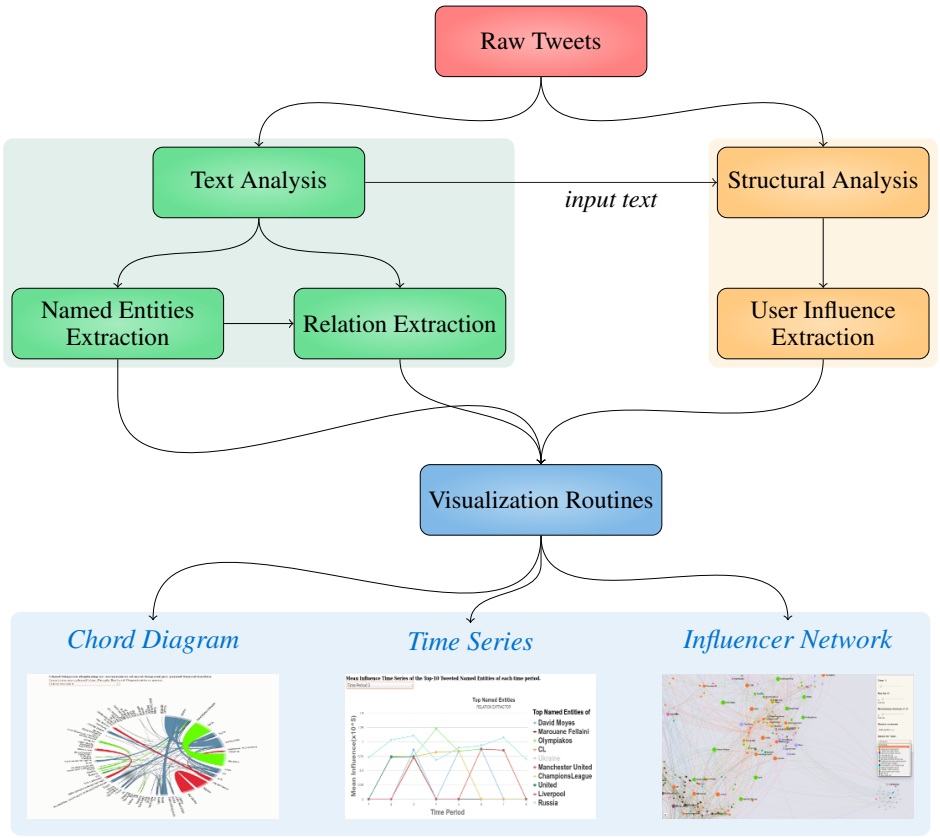


Figure 1: Proposed workflow diagram for visualizing trends in Twitter.

2.2 Text Analysis

The goal of text analysis is to tackle the task of "important" event extraction from the vast stream of Twitter data. To do so, we relied on an innovative work [10], that combines state-of-the-art methods and tools.

In particular, the outline of the pipeline used, is as follows:

1. *Preprocessing*: Cleaning the input data is crucial for accurate Named Entity Recognition and Relation Extraction. This is done through cleaning of the text from residual html tags, tokenization and user name resolution(replacing user mentions with their according twitter username, enabling us to find more relations).
2. *Named Entity Recognition*: Afterwards, we move on to find the named entities in each tweet, as well as their types. For this task, the Stanford Named Entity Recognizer(Stanford NER) [6] was used because it is reported [5] to achieve the highest average precision. We focus on named entities of type Location, Person and Organization that are much more probable to be part of "important" events, where "important" would indicate tweets containing informational value to the user, such as a politician attending a summit, or an organization making a press conference.
3. *Relation Extraction and Selection*: Subsequently, this module aims to extract meaningful relations of the form subject-predicate-object, with subject and object being among the extracted named entities. This is done using ClausIE [4], in conjunction with several modifications tailored for the task. Using the frequency of the occurrences of the named entities, we filter the extracted relations, in order to retrieve the most important news.

2.3 Structural Analysis

The core idea of this part of the work is to identify influential users on an on-line social community like Twitter. In detail, we implemented the novel method of Topic Sensitive-Supervised Random Walks(TS-SRW) [9]. This method utilizes structural information about the users/nodes and interactions/edges, as well as textual content affiliated to each node(the tweets of the user) in the network, to measure topic-sensitive influence of nodes.

To do so, a Latent Dirichlet Allocation [2] model is first used to extract the per topic distributions of each user. Then, the similarity between users based on those topic proportions is computed, as it is needed for the Supervised Random Walk [1]. Finally, exploiting the structure of the network to create weights between edges and the similarity values of the nodes

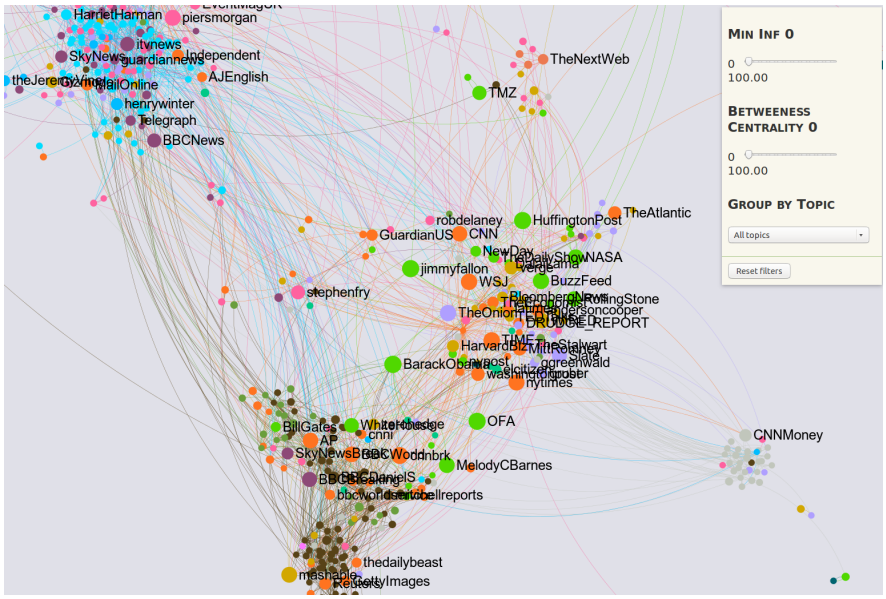


Figure 2: Example of a User Network

regarding topics, a Biased Random Walk is performed on the graph. The intuition is that the higher the topical similarity and the edge weight, the higher the transition probability will be, leading us to more influential nodes.
At the end of this procedure a PageRank-like score is computed for each node, denoting the influence of the user in the network.

3 Experiment and Visualization Tools

The previous stage of the workflow provides us with processed information that can now be combined and subsequently visualized, in order to gain insights about the data. There are many ways to use the information produced and here we will only point out a few of the visualization schemes that can augment the expressiveness of information, enhance user interaction and facilitate knowledge discovery.
In order to showcase the usefulness of the proposed workflow, we conducted an experiment applying it to the Snow 2014 Data Challenge test dataset[2]. This dataset consists of 1.089.909 tweets, by 560.009 users, resulting in 963.685 edges. The keywords used to gather the data, through the Twitter Streaming API, were *Syria*, *terror*, *Ukraine* and *bitcoin*.
As noted before, the main idea was to fuse the knowledge regarding the influence of the users with the relations of the named entities found in the tweets of those users. A few descriptive ways to convey facts about these complex interconnections of users, relations and user-influence are adumbrated below[3].

[2]publicly available at `http://figshare.com/articles/SNOW_2014_Data_Challenge/1003755`

[3]Visit `http://users.iit.demokritos.gr/~bogas.ko/` for live-interactive demos.

User Network

A multiple-aspect browsing of the data is possible through a network representation. In particular, we are interested in how the different users are connected with each other, how influential they are and what is their main topic of conversation. This can be visually expressed through a User Network. It is made feasible by utilizing the results of the Structural Analysis and is depicted in figure 2. Each node is a user, with radius proportional to their influence, color based on their main topic of interest and the links between them are interactions such as mentions, replies, retweets etc.

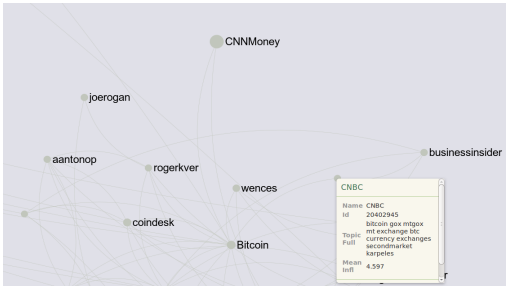


Figure 3: Network filtered by topic about finance.

This representation is very rich, allowing us to view the network from different perspectives. In an example scenario such as viral marketing or opinion propagation, one could be interested in finding the influential users regarding financial matters. We could filter the nodes according to a topic re-

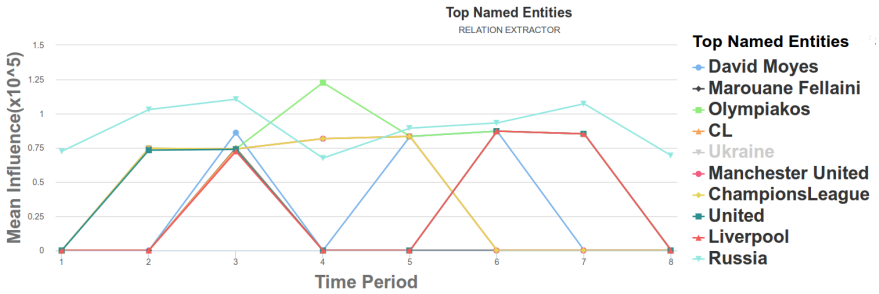


Figure 4: Time Series of the mean influence of named entities over 8 time periods.

garding finance and find the most influential users, based on their mean influence, or the users that act as connecting links between the different communities, based on their betweeness centrality [7]. An example is depicted in figure 3.
We have filtered the network to focus on users discussing about financial topics. It can be seen by the size of the nodes that *CNNMoney* is more influential in this community than *CNBC* or the *Bitcoin* profile. However, using the betweeness centrality filter, one could see that the *Bitcoin* profile is superior in terms of being the interconnecting node between different users.

This view of the users as interconnected nodes, along with filtering capabilities regarding topic, mean influence or other kinds of attributes for each user, allow for diverse and multi-purpose exploratory analysis. Others for example might be more interested in what topics are the most dominant in the network, what are the influential users talking about in a specific time period, which nodes are the main news providers, which users tend to create hubs around them and more.

Chord Diagram

A different view of the data can be offered through a Chord Diagram. A Chord Diagram is a way of exposing the inter-relationships of data [8]. In our case, we focused on the frequency of the named entities found in our corpus, as well, as the co-occurrences of the different pairs, across time. For this purpose, we calculated a co-occurrence matrix regarding all the named entities while taking into account the frequency of the named entities for different time periods. The resulting tool would create a Chord Diagram as the one shown in figure 5.

The insights gained from this type of visualization are multiple. For example, one can see over the different time periods which Named Entities were the most frequent. As expected for our example dataset, entities like Ukraine, Syria, Obama etc. are the most important ones. But looking at the diagrams, someone might be perplexed as to why Ukraine co-occurs so often with Venezuela. This unexpected link is explained by looking at some other top named entities such as *Liubov Yeremicheva*, *Yaryna Pochtarenko*, who are photographers that have taken pictures of Ukraine protesters

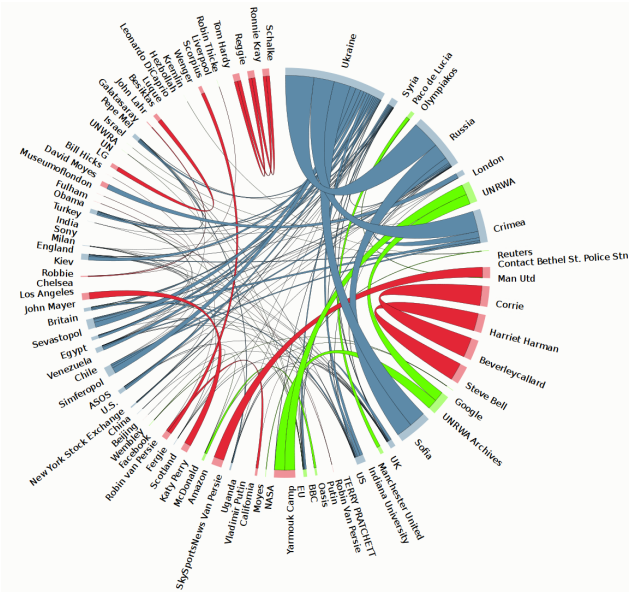


Figure 5: Example of a Chord Diagram

declaring support to their Venezuelan counterparts[4]. The Chord Diagram emphasized this connection, helping us unveil and understand the hidden link between the two events.

Time Series

A more conventional visualization scheme is a Time Series analysis of the influence of the named entities. Specifically, we focus on the top-10 most frequent named entities per time period and calculate the mean influence of each entity, according to the influence of the users that write about them. Then, we can portray how the influence of each term fluctuates between different time periods, as shown in figure 4.

Some entities have generally high influence over all time periods, such as *Russia*(light blue), others spike and disap-

[4]Protests in February, 2014 `http://ireport.cnn.com/docs/DOC-1097482`

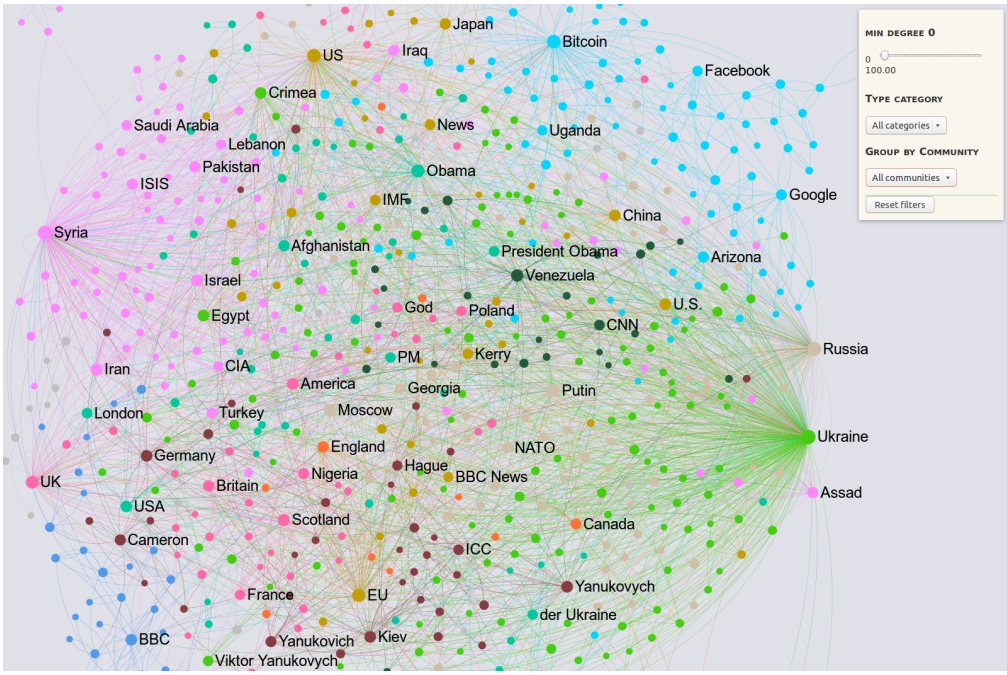


Figure 6: Example of Named Entities Network

pear, such as *Manchester United*(pink), *CL*(orange), because they were triggered by a specific event that ended (a football match).

These are very interesting in terms of finding trending keywords, events or what the topic of interest of the most influential users.

Named Entities Network

Another interesting visualization design displays the named entities found in the tweets as nodes in an interaction network. The connections between nodes denote the appearance of this pair of named entities in a tweet and the size of each node expresses the aggregated user influence of the profiles that talk about this named entity. Moreover, projecting the named entities as nodes in a plane and taking advantage of their connections allows us to find[3] communities of named entities, that is sets of named entities semantically close, based on co-occurrences. An instance of this network is shown in figure 6.

This network view at the granularity level of named entities enables the end user to understand the main topics of discussions through trending keywords at glance. Moreover, one can easily see what influential users talk about and how these named entities interlink in order to form topics of interest.Finally, one can use sophisticated filters in order to find meaningful interconnections between the entities. For example, one could filter based on the type of the named entities looking for connections of organizations or companies with persons, when investigating corruption news.

4 Conclusion and Future work

In this work we proposed a framework for analyzing and exploring twitter streams.This is done through analysis and fusion of information about the influence of the users, the topics of discussion, the co-occurrences of named entities in these topics and the interactions of the users. The results of this process are presented to the end user through expressive visualization tools, enabling users to draw useful conclusions about the data.

As future steps, we would like to incorporate more analysis tools in our framework, such as descriptive statistics regarding topics of discussion or users' interconnections. This, in accordance with new visualization tools, will provide richer information to the end users, such as ways of tracking the source of an event or the probability of it being a false rumor spread between users. Another possible future expansion would be to process the news in a streaming fashion, allowing journalists to monitor multiple story-lines at the same time. That is, journalists would define a few keywords they would like to focus on and using these tools they could delve into specific details about the diffusion of news, the evolution of a topic over time etc., gaining insights and drawing conclusions.

Acknowledgments

This work was supported by REVEAL[5] project, which has received funding by the European Unions 7th Framework Program for research, technology development and demonstration under the Grant Agreements No. FP7-610928.

[5] http://revealproject.eu/

References

[1] L. Backstrom and J. Leskovec. Supervised random walks: Predicting and recommending links in social networks. In *Proceedings of the 4th ACM WSDM Conference*, New York, NY, USA, 2011.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3, March 2003.

[3] Vincent D Blondel, Jean loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks, 2008.

[4] Luciano Del Corro and Rainer Gemulla. Clausie: Clause-based open information extraction. In *Proceedings of the 22nd WWW Conference*, New York, NY, USA, 2013.

[5] L. Derczynski, D. Maynard, G. Rizzo, M. van Erp, G. Gorrell, R. Troncy, J. Petrak, and K. Bontcheva. Analysis of named entity recognition and linking for tweets. *Inf. Process. Manage.*, 51(2), 2015.

[6] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on ACL*, 2005.

[7] Linton C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 1977.

[8] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5), 2006.

[9] G. Katsimpras, D.s Vogiatzis, and G. Paliouras. Determining influential users with supervised random walks. In *Proceedings of the 24th WWW Conference*, New York, NY, USA, 2015.

[10] G Katsios, S Vakulenko, A Krithara, and G Paliouras. Towards open domain event extraction from twitter: Revealing entity relations. In *Proceedings of the 4th DeRiVE workshop, 12th ESWC 2015, Slovenia*, 2015.

Labeled Topics for News Corpora Using Word Embeddings and Keyword Identification

Abdulkareem Alsudais, Hovig Tchalian, Brian Hilton
Claremont Graduate University
{abdulkareem.alsudais, hovig.tchalian, brian.hilton}@cgu.edu

Abstract

In this paper, a simple pipeline for identifying labeled topics for temporally ordered and topic-specific news corpora using word embeddings and keyword identification is proposed. The steps in the pipeline rely on NLP techniques to identify keywords, corpus periodization, and word embeddings. The proposed method is evaluated by applying it on a dataset consisting of TV and Radio transcripts on "Donald Trump." The results demonstrated that the topics captured using this pipeline are more coherent and informative than ones generated using Latent Dirichlet Allocation. Findings from this preliminary experiment suggest that word embeddings models and common NLP keyword identification techniques can be used to identify coherent and labeled topics for a temporally ordered news corpus.

1 Introduction

"Word embeddings" refer to models that create dense vector representation for words or phrases in a text corpus by utilizing their immediate syntactic context, defined by a window with proximate terms. Recently, these models have gained popularity, partially due to advances in computing powers that have made creating vectors for large corpora more feasible [Goth, 2016]. Successful models for generating vector representations for words and phrases such as word2vec and its SKIPGRAM model [Mikolov *et al.*, 2013a], GloVe [Pnnington *et al.*, 2014], Swivel [Shazeer *et al.*, 2016] and others [Levy and Goldberg, 2014b; Turian *et al.*, 2010] have proven to be successful in performing various language-related tasks. These tasks include solving analogies equations and generating word similarities.

Researchers and scientists in Natural Language Processing (NLP) have leveraged these word embeddings models to solve various research problems related to text corpora. For example, Mikolov *et al.,* [2013b] used them to translate texts between English and Spanish; Alemi and Ginsparg [2015] used them to segment text documents; Lebret *et al.,* [2015] used them to generate image captions; and Leeuwenberga *et al,.*[2016] used them to find synonyms for words. These examples demonstrate the effectiveness of solutions that rely on word and phrase vectors as produced by the aforementioned word embeddings models. The success of these approaches also highlights the potential of word embeddings models for solving common and current NLP and text mining related research problems. In this paper, the ability to utilize word embeddings models and keyword identification techniques to generate labeled topics for news corpora is investigated.

One common feature of word embeddings models is a function that identifies a list of words or phrases that are most similar to a given word. This function simply locates the words or phrases that have similar vectors to a given word. According to Mikolov *et al.*, [2013c], the types of similarities returned by the function in SKIPGRAM varies, while Levy and Goldberg [2014a] suggest that the similarities are mostly topical.

While representing a significant step forward, current word embeddings tools and models nonetheless still require customization, modification, and enhancement in order to accomplish domain-specific and NLP-related tasks. Therefore, researchers studying and examining text corpora might benefit from learning how other researches created pipelines or blueprints that utilize word embeddings models to solve and complete specific problems. Additionally, new solutions that leverage recent advances in NLP to create novel applications of word embeddings tools have the potential to advance the NLP field even further.

Most word embeddings methods, for instance, rely only on the linear context of the text, which results in losing additional contextual information present in the text. Levy and Goldberg [2014a] proposed a new word embeddings method that generalizes SKIPGRAM by preserving dependencies such as "direct object" or "nominal subject" that each word in the text represents. They argued that incorporating these dependencies improved the similarity results generated by SKIPGRAM. We build on Levy and Goldberg research by demonstrating how adding additional contextual markers can enhance the performance of specific tasks popular when using word embeddings models.

One particularly appropriate application of such enhanced solutions is discovering topics of a text corpus, and specifically in the area of topic modeling, an algorithmic approach that generates thematic clusters in a corpus based on the distribution of co-occurrence probabilities across word vectors. Research in topic modeling was pioneered by the work of Blei *et al.*,[2003] and their LDA (Latent Dirichlet Allocation)

method. There are a number of solutions that have extended the LDA method, the most successful and widely accepted of which is Labeled-LDA [Ramage *et al.*, 2009]. Labeled-LDA extend LDA and generate not only topics for the corpus, but also labels that define the topics. The algorithm leverages the labels or tags linked to each document in the corpus. For instance, if the examined corpus consists of academic papers on text mining, utilizing the authors' keywords for each paper, the algorithm generates labeled topics for the corpus, such as "entity resolution," and "deep learning."

The growing work in topic modeling represents precisely the kind of opportunity for the enhanced NLP applications we discussed above. When analyzing text corpora, researchers commonly use topic modeling algorithms to generate topics latent in a text corpus. These topics can be used to establish context for the corpus, and to identify, efficiently and effectively, the most important themes in the corpus. To the best of our knowledge, there has not been any work on capturing labeled topics for a corpus using word embeddings.

In this paper, the topical similarities as identified by SKIPGRAM are leveraged to generate corpus-wide labeled topics. We find that utilizing the similarities as generated by Mikolov *et al.,* [2013a] to generate labeled topics offers a significant improvement over approaches that do not incorporate word embeddings to generate topics. The proposed pipeline relies on recent advances in temporal text mining an widely accepted Natural Language Processing techniques such as tokenizing, lemmatizing, Part of Speech (POS) tagging, and keyword identification to generate "context" for the corpus. We argue that by providing "context" for the corpus, SKIPGRAM can be used to generate labeled topics that are more informative than ones generated by LDA, in particular for news corpora. Our application demonstrates that word embeddings can be effective and informative in generating labeled topics for text corpora when used to supplement common NLP techniques performed on the corpus.

2 Methodology

In this section, the steps in the proposed pipeline are briefly described. Figure 1 illustrates the entire pipeline, including the results achieved after each step.

2.1 Periodization

Corpus periodization is the process of segmenting a corpus into a set of smaller and discursively coherent periods while retaining the chronological order of the corpus. Social scientists in fields such as sociology and history commonly use periodization to study various changes in a specific discourse across time by fragmenting a corpus into a set of focal periods [Morley and Bayley, 2009] . For example, when Ruef [1999] examined thirty years of textual news articles on market reform in the U.S. healthcare sector, he identified historical events and political acts in this time frame, and used them to segment the corpus into periods. Corpus periodization is incorporated in the proposed pipeline and utilized to identify the most important keyowords or noun phrases of the corpus. We explain the periodization method and the justification for using it in another paper [Alsudais and Tchalian, 2016].

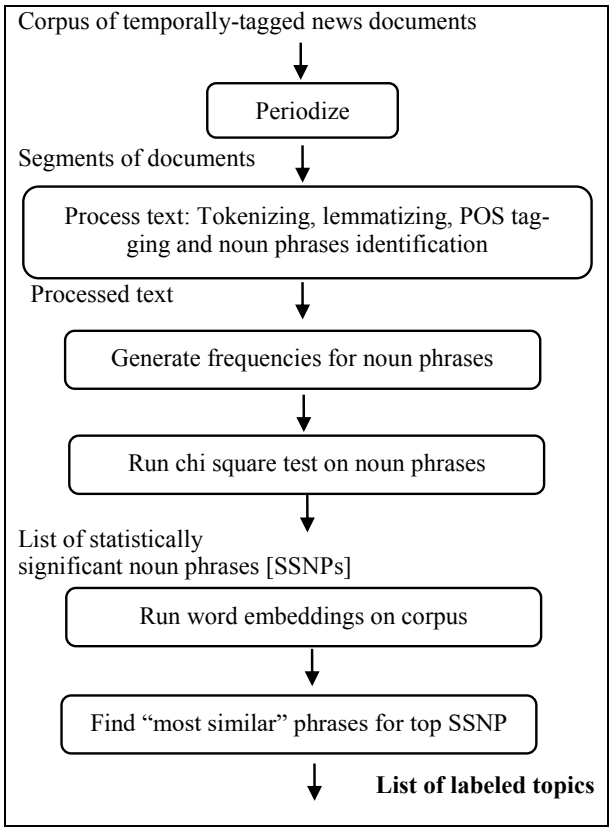


Figure 1. The steps in the propsed pipeline

2.2 Text Preprocesing

In this step, a number of common text processing techniques are applied on the corpus. The purpose of this step is to refine the corpus and only retain information relevant to the process of topic labeling and generation. First, all the news articles in each period are tokenized into a set of sentences. Then, all words in the sentences are tokenized, transformed to lower case, and subsequently lemmatized, the accepted approach within the NLP field. By changing words to lower case and lemmatizing them, natural synonyms such as "Companies" and "company" get combined and counted as the same entity.

2.3 Noun Phrases Extraction

The purpose of this step is to identify the noun phrases most central to the corpus, a critical intermediate step in this pipeline. To complete this step, all the noun phrases in each period are captured and their frequencies are counted. Afterwards, a chi square goodness of fit test is computed with the frequency counts of the noun phrases in each period as the columns. The result of this step is a list of chi square values for each unique noun phrase in the corpus. A list of statically significant noun phrases is then created according to a selected significance level or threshold. The list includes all the noun phrases that pass the defined threshold level.

2.4 Word Embeddings

After preprocessing the corpus, removing all non-noun phrases, and saving noun phrases in the corpus as single units, in this step, a word embeddings model is run on the modified corpus. The result of this step is a dense vector representation for the noun phrases in the text. SKIPGRAM is used in this paper to generate the vector representations for the noun phrases. Since various word embeddings models can produce different vectors for the same word or phrase in a corpus, using a different word embeddings model may not result in generating topical similarities identical to the ones SKIPGRAM generates.

2.5 Labeled Topics Generation

The final step in this pipeline is to produce labeled topics for the corpus. Using the "most similar" function in word2vec, lists of the most similar noun phrases are generated for all the statistically significant noun phrases (SSNP). This function simply calculates and detects the noun phrases that have vectors that are most similar to a given word or phrase. The SSNPs are then used as labels for the generated topics. While only topics for the top statically significant noun phrases were captured for the purposes of this paper, further extensions of this work will also identify topics for all SSNPs and combine similar and overlapping ones to create multi-labeled topics that fully capture the all the most popular themes in the corpus.

3 Experiment

In this section, the results of applying the proposed method on a corpus of temporally tagged news corpora are demonstrated.

3.1 Dataset

In this paper, a dataset consisting of transcripts of news-related television and radio shows in which the name "Donald Trump" appeared is used. This dataset is selected to demonstrate the effectiveness of the method proposed in this paper in generating labeled topics for temporally ordered and topic-specific news corpora.

Donald Trump announced his candidacy for President of the United States on June 16, 2015. Ever since, a considerable public dialogue has been taking place around the various 'hot button' topics his candidacy has elicited. Whatever the merit of the dialogue itself, its time-bound nature, considerable volume and focus on a limited number of topics makes it an ideal dataset for purposes of evaluating the method proposed in this paper. In particular, the method can be evaluated by measuring its accuracy in capturing and labeling the topics latent in the news corpus representing the political dialogue.

Two sources were used for the dataset: Fox News Network and National Public Radio (NPR), which together provide a substantial sample of the range of political discussions surrounding the candidate. The dataset includes transcripts of shows that aired from June 1st, 2015 to March 31st, 2016. The total number of news transcripts is 2,271 documents. Just over a third of the corpus, 1,528 documents, aired on Fox News Network while the rest, 743 documents, aired on NPR.

3.2 Periodization

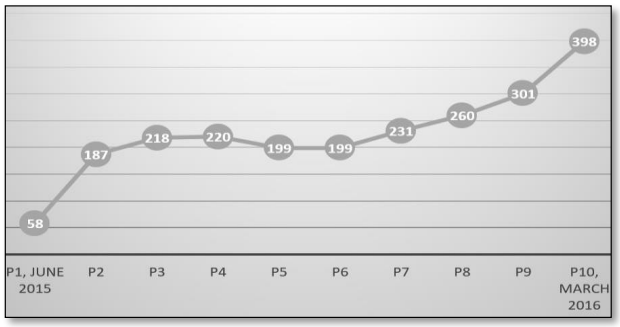


Figure 2. The number of articles in each period.

When the number of the initial temporal break points in the corpus is small, using a corpus periodization method to segment the corpus is not necessary. In this dataset, the initial points were the individual months between June, 2015 and March 2016. Due to the small number of the initial points, natural break points were used in this experiment and articles were grouped according to the month and year combination of when they were published. Accordingly, a set of ten periods was created. Figure 2 shows the breakdown of the periods, and the number of articles in each one.

Rank	Keyword	Rank	Keyword
1	Delegate	13	Cleveland
2	Scott walker	14	Cruz
3	Convention	15	Caucus
4	Paris	16	Iowa
5	Biden	17	South carolina
6	Carly Fiorina	18	Assad
7	Muslims	19	Nevada
8	Iowa caucuse	20	Iowa and new hampshire
9	Illegal immigrant	21	Committee
10	Carson	22	Planned Parenthood
11	Server	23	Putin
12	Home state	24	Iran
25	Caucuse	37	Gun
26	New Hampshire	38	Primary
27	Kasich	39	Syria
28	Murder	40	Paul ryan
29	Mexico	41	Shooting
30	Terrorism	42	Nuclear Weapon
31	Sanders	43	Russia
32	Refugee	44	Afghanistan
33	Illegal immigration	45	Agreement
34	Jeb Bush	46	Nuclear deal
35	Michigan	47	Sanction
36	Benghazi	48	Outsider

Table 1. Top SSNPs, as captured by periodization, the chi-square test and traditional bag-of-words approaches

3.3 Noun Phrases / Keywords

After preprocessing the corpus using the techniques described in section 2.2, all noun phrases were identified along with the number of times they were used in each period. Accordingly, the chi square values were computed for each noun phrase. Finally, a list of Statistically Significant Noun Phrases (SSNP) was generated based on the chi square values of the noun phrases at a significance level of 95%. The list contained over 700 noun phrases. In table 1, the top 48 SSNPs are listed. These words and phrases are later used as labels that define the topics. This list alone provides some contexts and summary of the corpus and the main themes discussed in the processed 2,271 documents across the ten periods.

3.4 Labeled topics

After identifying the list of statistically significant noun phrases, all the documents in the dataset were processed using word2vec to create dense vector representations for all the unique phrases in the corpus. The final step in the pipeline was completed by simply querying word2vec to retrieve the lists of the most similar word or phrases to each of the words and phrases in the SSNPs list. These lists are used as topics, and the SSNP used to create them are used as the labels for the topics.

We observed that labeled topics are not as informative when the label or noun phrase is a proper noun, such as "Biden" or "Cruz." For example, for the label "Cruz," the topic contained the names of a number of other presidential candidates such Rubio and Bush. Thus, more rigorous testing is needed to implement methodical changes that systematically, and perhaps iteratively, refine the topics. Additionally, there are instances where the labeled topics overlap and share the same underlying terms. Therefore, identifying and combining these topics is needed.

In table 2, a sample of the generated topics is displayed with more emphasis on topics that are not labeled with a proper noun. The effectiveness of the performance of the method proposed in this paper is demonstrated by comparing the generated labeled topics to topics generated by LDA on the same corpus. Some topics such "Planned parenthood" and "Terrorism" are more coherent than others.

Selected Labeled Topics Generated by the Method Proposed in this Paper		**Topics as Generaetd by LDA**
Label	**Topic**	
Delegate	['ballot', 'convention', 'primary', 'first ballot', '1,237 delegate', '30 percent', '50 percent', 'republican primary', '20 percent', 'contested convention']	'-- ', 'bolling', 'williams', 'guilfoyle ', 'gutfeld ', 'perino: ', 'that's', 'video', 'right', 'clip'
Scott walker	['rick santorum', 'john kasich', 'rick perry', 'mike huckabee', 'chris christie', 'rand paul', 'lindsey graham', 'mitt romney', 'marco rubio', 'jeb bush']	'trump ', 'donald ', 'cruz ', 'ted ', 'republican ', "he's ", 'hillary ', 'rubio ', 'win ', 'campaign'
Paris	['brussels', 'belgium', 'paris attack', 'terror attack', 'france', 'san bernardino', 'mali', 'turkey', 'isis terrorist', 'bombing']	'kelly', '-- ', 'so,', 'unidentified', 'video', 'clip', 'know', 'male', 'well, ', "he's"
Muslims	['complete shutdown', 'more muslims', 'temporary ban', 'southern border', 'christians', 'refugee', 'proposed ban', 'jihad', 'total and complete shutdown', 'fear']	'carlson', 'planned ', 'abortion', 'parenthood', 'pro-life', 'body', 'fields', 'parenthood', 'rose ', 'abortion'
Iowa caucuse	['super tuesday', 'new hampshire primary', 'iowa caucus', 'new poll', 'caucuse', 'republican race', 'national poll', 'south carolina primary', 'next week', 'tuesday']	"o'reilly", '-- ', 'unidentified', 'right', 'so, ', 'yes', "that's", 'watters:', 'male', 'clip'
Illegal immigrant	['illegal alien', 'san francisco', 'deportation', 'five time', 'criminal', 'sanctuary city', 'immigrant', 'criminal alien', 'felon', 'citizen']	'-- ', 'wallace', "he's", 'well', "that's", 'debate', 'know', 'republican', 'trump', 'lot'
Server	['e-mail', 'classified information', 'email', 'private server', 'mail', 'top secret', 'state department', 'e', 'private e-mail', 'benghazi']	'siegel', 'robert', 'greene', 'know', 'well', "you're", 'kind', "that's", 'mean', 'young'
Home state	['double digit', 'third place', 'florida', 'second place', 'other state', 'latest poll', 'win', 'ohio and florida', 'winner', 'wisconsin']	'baier', 'president', 'fox', 'news', 'video', '-- ', 'end', 'begin ', 'clip', 'u.s.'
Caucus	['early state', 'caucuse', 'ground game', 'turnout', 'republican primary', 'voting', 'polling', 'iowa and new hampshire', 'primary', 'evangelical']	'trump', 'kurtz', 'recording', 'media', 'donald', 'ari', "he's", 'unidentified', 'press ', 'news\n'
Terrorism	['threat', 'terror', 'homeland', 'radical islam', 'initial response', 'middle east', 'al qaeda', 'islamic state', 'isil', 'region']	'pope ', 'religious', 'carson', 'ben', 'faith', 'christian', 'church', 'catholic', 'joe', 'pope.'
Assad	['vacuum', 'putin', 'coalition', 'red line', 'ukraine', 'vladimir putin', 'force', 'no-fly zone', 'iraq', 'ally']	'trump', '-- ', 'know', 'donald', "we're ", '[applause] ', 'great ', 'it.', 'trump', 'lot'
Planned parenthood	['abortion', 'body part', 'federal funding', 'abortion part', 'entire federal government', 'funding', 'taxpayer funding', 'abortion practice', 'reimbursement', 'aborted fetuse']	'audie', 'cornish', 'scott', 'scott:', 'horsley', 'e.', 'meyers', 'e.j.', 'as', 'dionne'
Committee	['hearing', 'inspector general', 'benghazi committee', 'classified information', 'justice department', 'document', 'email', 'testimony', 'state department', 'mrs. clinton']	'cavuto: ', 'it`s ', '-- ', 'don`t ', 'well, ', 'i`m ', 'that`s ', 'he`s ', 'we`re ', 'so,\n'
Iran	['sanction', 'agreement', 'nuclear weapon', 'nuclear deal', 'north korea', 'iranians', 'deal', 'u.n.', 'nuclear program', 'regime']	'- ', '(soundbite ', 'david ', 'archived ', 'steve ', 'gonyea: ', 'inskeep: ', 'sarah ', "npr's ", 'donald\n'

Table 2. Summary of selected labeled topics as generated by proposed pipeline (left) and topics as generated by LDA.

4. Conclusion and Future Work

In this paper, a simple pipeline for identifying labeled topics for temporally ordered and topic-specific news corpora is proposed. The steps in this pipeline rely on 1) widely accepted Natural Language Processing techniques and approaches that preprocess and clean documents in the corpus, 2) a periodization method that utilizes the temporal features of the corpus to periodize it and create comprehensive textual keywords, and 3) a word embeddings model that creates dense vector representation for all unique words and phrases in the corpus.

The main contributions of this paper are 1) a demonstration that word embeddings models can be used to identify coherent topics for a corpus and 2) a pipeline that can be replicated to produce a list of labeled topics for a temporally ordered news corpus.

The effectiveness of this proposed method was demonstrated by applying it to a corpus consisting of 2,271 television and radio transcripts containing the term "Donald Trump." We demonstrated that the labeled topics generated by the proposed method were more informative than ones generated by Latent Dirichlet Allocation (LDA). Our method captured more coherent topics than ones generated by LDA. Furthermore, this method more accurately captured semantic and syntactic context, as confirmed in the topics labeled "Planned Parenthood" and "Terrorism."

Additional and more rigorous testing and evaluation of this method is necessary. For instance, it is common to preprocess a corpus before generating topics with LDA. We acknowledge that, in this paper, LDA was run on the corpus without applying any preprocessing on the text. Preprocessing the corpus might produce topics that are more concise and relevant. Thus, it is important to run additional tests, in order to determine whether such preprocessing can improve the accuracy, coherence and relevance of the standard LDA approach, possibly helping explain a proportion of the contribution difference between that method and the one proposed in this paper. Very minor discrepancies in the topics generated using the processed and un-processed runs of this method proposed here strongly suggest that the difference would be negligible. Additionally, topic coherence measures such as C_V and C_P [Röder *et al.*, 2015] should be used to evaluate the results and quantitatively asses the coherence of each individual topic.

This work can also be refined and extended in several ways. For example, examining the labeled topics reveals clear overlap between some of the topics. There are opportunities, therefore, to further leverage the similarities between the terms and topics as identified by the word embeddings model, in order to merge and collapse certain topics. The results of this should be a set of topics that provide a comprehensive and complete summary for the corpus being studied. Moreover, examining the labeled topics also reveals different types and classes of topics. Thus, future work includes examining these types and creating a method that systematically classify them.

References

[Alemi and Ginsparg, 2015] Alexander A Alemi and Paul Ginsparg. Text Segmentation based on Semantic Word Embeddings. *arXiv Preprint*. arXiv:1503.05543.

[Alsudais and Tchalian, 2016] Abdulkareem Alsudais and Hovig Tchalian, H. Corpus Periodization Framework to Periodize a Temporally Ordered Text Corpus. In *Proceedings of the 22nd Americas Conference on Information Systems (AMCIS)*.

[Blei *et al.*, 2003] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3, 993–1022.

[Goth, 2016] Gregory Goth. Deep or Shallow, NLP Is Breaking Out. *Communication of the ACM*, 59(3), 13–16.

[Lebret *et al.*, 2015] Remi Lebret, Pedro O. Pinheiro, and Ronan Collobert. Phrase-based Image Captioning. *arXiv Preprint.* arXiv:1502.03671.

[Leeuwenberga *et al.*, 2016] Artuur Leeuwenberga, Mihaela Velab, Jon Dehdaribc, and Josef van Genabithb. A Minimally Supervised Approach for Synonym Extraction with Word Embeddings. *The Prague Bulletin of Mathematical Linguistics*, (105), 111–142.

[Levy and Goldberg, 2014a] Omer Levy and Yoav Goldberg. Dependency-Based Word Embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers)*. pp. 302–308. Baltimore, Maryland, USA: Association for Computational Linguistics.

[Levy and Goldberg, 2014b] Omer Levy and Yoav Goldberg. Neural Word Embedding as Implicit Matrix Factorization. A*dvances in Neural Information Processing Systems*, 2177–2185.

[Mikolov *et al.*, 2013a] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. *Advances in Neural Information Processing Systems*, 26, 3111–3119.

[Mikolov *et al.*, 2013b] Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *arXiv Preprint*. arXiv:1301.3781.

[Mikolov *et al.*, 2013c] Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. Exploiting Similarities among Languages for Machine Translation. *arXiv Preprint*.

[Morley and Bayley, 2009] John Morley and Paul Bayley (Eds). Corpus-assisted discourse studies on the Iraq conflict: Wording the war. Routledge.

[Pnnington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe : Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pp. 1532–1543. Association for Computational Linguistics.

[Ramage *et al.*, 2009] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. Labeled LDA : A supervised topic model for credit attribution in multi-labeled corpora. In *EMNLP '09 Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. pp. 248–256. Association for Computational Linguistics.

[Röder *et al.*, 2015] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the Space of Topic Coherence Measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*.

[Ruef, 1999] Martin Ruef. Social ontology and the dynamics of organizational forms: Creating market actors in the healthcare feld, 1966-1994. *Social Forces*, 77(4), 1403–1432. 1999.

[Shazeer *et al.*, 2016] Noam Shazeer, Ryan Doherty, Colin Evans, and Chris Waterson. Swivel: Improving Embeddings by Noticing What's Missing. *arXiv Preprint*. 2016.

[Turian *et al.*, 2010] Joseph Turian, Lev Ratinov, and Yoshua Bengio. Word representations : A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. pp. 384–394.

Diachronic Evaluation of Newspapers Language between Different Idioms

Daniela Gîfu

Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași, Romania
daniela.gifu@info.uaic.ro

Abstract

Due to various reasons, it is not rare that two cognate languages become strained for a period of time, only to become closer for another period of time. Traditionally the degree of similarity was assets by linguistics on the basis of their expertise. However, it is hardly possible to cover a large material only by human effort. We present a methodology of diachronic investigation on news corpora which determines the degree of similarity between cognate languages.

1 Introduction

The present work investigates the linguistic crisis that affects the journalistic laguage in two countries, Romania (including three historical regions: Moldavua, Tranylvania and Wallachia), and the Republic of Moldavia (known as Bessarabia), which until the early 19th century were one state. This linguistic contrastive study between Romania and Bessarabia allows intercepting many similarities, especially in diachrony. The similarities of the Romance languages are becoming more numerous, as we descend deeper into past. [Densuianu, 1902]. Other important differences were also detected, perhaps due to the influence of Russian language reflected on the Bessarabian language, starting from the middle of the 19th century. It is also important to note that starting with the 19th century the Romanian language was influenced for more than 30% by French and Italian (two Romance languages as Romanian). We analyse, via automatic corpus methodology, the similarity of the two languages, between two periods – before the Second World War and after the fall of communist regime.

The methodology we present is language independent and it can be applied to any two corpora, let's call them target and source. In a nutshell, we first determine the characteristics of each of the four corpora and then we compute the similarity of pairs extracted from target and source corpora, on the basis of these characteristics. We take into account all levels of linguistics analysis in order to derive the language characteristics of a language: lexical, morphological, syntactical, semantically and discourse level respectively. We use a large suite of statistical methods in order to determine.

The similarity considering both words, via word embedding techniques and topics, via LDA type analysis. The methodology we present is offers a basis for future large-scale studies, having a large impact on reducing the amount of human effort required by socio-historical linguistic analysis of language idioms in general.

The results of this contrastive analysis highlight the significant changes in the distribution of terms that best reflects the differences in writing style, ranging from sentence and paragraph structure, to topic cohesion. Finally, a formula computes the similarity in a complete and objective way.

In order to meaningfully carry out this analysis we compiled a corpus of journal articles from the geo-political distinct cognates: Romanian and Bessarabian. A large corpus (over 2.6 million lexical tokens), chronologically ordered since the second decade of the 19th century (1817-2015), was developed, structured in four independent collections of publications corresponding to Moldavia – 68373 words, Wallachia – 143612, Transylvania – 2294108 words, and Bessarabia – 92499 words. Based on this corpus we explore the diachronic phenomenon in order to identify statistically Romanian epochs reflected on the printing press and linguistic similarities from Bessarabian press. The Republic of Moldavia was a part of Romania (including three Moldavia, Wallachia, Transylvania) until 1812, and then from 1918 to 1941, becoming an independent state after 1991.

These texts can form the basis of an analytic process that aims to capture the semi-automatic deviations from the current norm. The automatically investigation offers a solution for historian as well, and historical significant correlation in the word usage may be discovered. In fact, diachronic analysis of cognate languages provides clues and insights into what the society considered adequate responses to social problems at a given moment. The rest of the paper is organized as follow: section 2 presents a brief review of relevant literature, section 3 depicts the corpora in details and the methodology, section 4 describes the analyse and interprets the results. Finally, the survey conclusions and future work are given in section 5.

2 Related Work

Many previous works [Leech et al., 2009: Davies, 2013] have focused mainly on the linguistic interpretation of the statistically results. Their hypotheses were based on the ways language chnages without considering their causes.

It has been established that some genetically related languages have a high degree of similarity to each other [Gooskens, 2006; Gooskens et al., 2008]. Various aspects present relevance when investigating the level of relatedness between languages, for example orthographic, phonetic, syntactic and semantic differences. The phonetic alterations have an orthographic correspondent, thus an alphabetic character correspondences [Delmestri and Cristianini, 2010].

The diachronically comparative studies of the Romance languages expose the presence of many similarities [Densuianu, 1902]. Latin language, the origin of Romanian, French, Italian, Portuguese, Spanish, was the starting point, but issues about substratum, superstratum and adstratum which contributed to differentiate languages were not set aside.

The development and use of software for natural language processing (NLP) highlight the defining aspects of the Romanian printing press (morphological and syntactic analysis, semantic analysis and, more recently, pragmatic analysis) that have many similarities to that of Bessarabia on the time axis that we have chosen. The rich literature tells its own story regarding the usefulness of technology and information services [Carstensen et al., 2009; Jurafsky & Martin, 2009; Manning & Schütze, 1999; Cole et al., 1998; Tufiş & Filip, 2002; Cristea & Butnariu, 2004; Trandabăţ et al., 2012, Popescu & Strapparava, 2013, 2014, Gîfu, 2015].

Until now, the Romanian diachronic phenomenon was analysed using various methods. One of them relies on the comparison of writing styles according to various indices: text features [Gîfu et al., 2016], textual formality [Eggins and Martin, 1997], and textual styles [Biber, 1987]. Another one is based on machine learning approach to explore the patterns that govern the lexical differences between two lexicons [Gîfu & Simionescu, 2016].

3 Corpus

A large corpus (over 2.6 millions lexical tokens and 6500 pages), chronologically ordered, since the second decade of the 19th century, was developed, structured in four independent collections of publications corresponding to Moldavia (68373 lexical tokens), Wallachia (143612 lexical tokens), Transylvania (2294108 lexical tokens), and Bessarabia (92499 lexical tokens) (see Table 1 for descriptive statistics).

Nowadays the first three regions form Romania, and Bessarabia was a part of Romania until 1812 and then from 1918 to 1941, becoming an independent state after 1991.

Region	Period	Total lexical tokens	Sources
Bessarabia	1817-2015	92499	Basarabia reînoită; Curierul; Candela; Deşteptarea; Viaţa economică din Bălţi; Solidaritatea; Ehos; Buletinul Arhiepiscopiei Chişinăului; Cuvânt moldovenesc; Ardealul; Basarabia; România nouă; Sfatul ţării; Democratul Basarabiei; Glasul Basarabiei; Luminătorul; Dreptatea; Basarabia Chişinăului; Literatura şi artă; Moldova Socialistă; Jurnal; Contrafort; Jurnal de Chişinău; Moldova suverană; Ziarul de gardă.
Moldavia	1829-2015	68373	Albina românească; Convorbiri literare; Curierul. Foaia intereselor generale; Constitutionalul; Moldova Socialistă; Scânteia; Noutatea; Deşteptarea; Bună ziua, Iaşi; Ziarul de Vrancea; Monitorul de Vaslui; Evenimentul regional al Moldovei; Imparţial.
Transylvania	1829-2015	2294108	Organulu Luminarei; Gazeta de Transilvania; Gazeta Transilvaniei; Telegrafulu Românu; Foaia pentru Minte Anima şi Literatură; Telegraful român; Transilvania; Federaţiunea; Gura Satului; Albina; Telegraful Românu; Familia; Aradu; Patria; Chemarea tinerimei române; Dreptatea; Aradul; Curierul creştin; Vatra

			românească; Echinox; Adevărul de Cluj; Făclia; Monitorul de Cluj; Bihoreanul.
Wallachia	1847-2015	43612	Curier românesc; Buletin. Gazeta oficială; România; Curierul românesc; Pressa, România liberă; Românulu; Timpul; Literatorul; Albina; Deşteptarea. Foaie pentru popor; Adeverul; Curierul artelor; Dimineaţa; Universul; Viitorul; Curentul; Universul literar; Adevărul; Adevărul literar şi artistic; Scânteia; Romania literară; Dimineaţa copiilor; Evenimentul zilei; Gândul; Ziua; Ziua news; Ziua veche;

Table 1. General corpus statistics

In other words, we talk about four Romanian idioms, covering two linguistic registers (journalistic, literature). To each text the following identification information are assigned (regions, year, publication, author).

It is also important that this corpus represents a first iteration towards building a Gold corpus for each region, centered on diachronic meta-annotation. It was prepared during 2 years. First, the corpus was edited in PDF, so we applied the boiling-plate technology to obtain raw text in TXT format (UTF-8 encoding), using Java PDF Library - Apache PDFBox. Then several corrections were made on the raw texts. Second, the processing phase continues with: segmentation, tokenization, lemmatization, part-of-speech, and NotInDict Markup using the UAIC POS-Tagger [Simionescu, 2011].

The result of the processing stage is an XML file that will be forwarded for other data processing. Moreover, we apply GGS grammar rules over the previous file. The GGS rules practically help to the disambiguation of the hyphen. In other words, one can understand when it is about hyphenation at the end of a row and when it deals with the components of the structure of certain words.

4 Methodology

We build diachronic vectors from corpus for each word, keeping on each slot the number of occurrences for a specific year. There are two variants of these vectors that we build, depending on whether different ortho-lexical realizations of the same word are considered the same, thus they count as one vector, or they lead to distinct vectors. The lexical vectors are relevant in time classification tasks, but less useful for topic identification. Consequently, we use one or the other set depending on the task that we need to resolve.

A snap-shot from a typical vector looks like:

```
768 pace /(EN) peace 1 1865 1 1868 17
1877 15 1878 3 1880 1 1897 4 1900
```

768 represents the total number of occurrences in the whole corpus, “pace”, Romania for peace, is the word and the occurrences of this word precedes the year. In this particular case, is easy to spot a variation in the period of 1877 and 1878, which, not incidentally, corresponds to an independence war fought exactly in those years. These types of non-random variances represent the basis for a diachronic analysis. In fact, each epoch is determined by a certain distribution of words.

As some topics of interest change over the time, the distribution of words in newspaper reflects this phenomenon accurately. Thus, by employing a suite of statistical test we can determine no-random changes in the word distribution. In [Popescu & Strapparava, 2013, 2014] was showed that there are a short period of few years within each many words change their distribution. As such, this specific period represents a transitional buffer between epochs. To determine the buffer period we apply to the from year to year. In particular we used three non parametric tests: Welch, run and ratio test.

We test respectively whether two samples come from the same statistical population, or whether there is a large variance with respect to the mean, or the ratio of change from year to year shown an upward or a downward trend.

For a very large corpus, like Google books for example, one can chose an arbitrary set of topics to investigate, but in this case we have a limited amount of data. Thus, we need first to indentify the topics that are represented in our corpus. For this we apply the LDA algorithm. At this step we use the non photo-lexical vectors are used. We filtered out set 25 topics the following topics for the target corpus, i.e. Romanian, like:

```
război, literatură, partide, stat,
pământ, muncitor, artă, sat, partidă /
(EN) war, literature, parties, state,
land, worker, art, village, party
```

For these topics the following epochs have been identified:

1832-1856	1920-1940
1856-1877	1940-1980
1877-1912	1980-1990
1912-1920	1990-2015

Table 2. Romanian Epochs in Newspapers

Considering this epochs as categories we build an SVM classificatory over whole target corpus (Weka implementation). We classified each news from the source corpus, i.e. Bessarabian. First thing we wanted to check was whether the classification is able to pin point correctly the source news. This will give a fairly accurate indication whether there is indeed a similarity over the epochs between the two cognate languages, or the model will assign a more or less random epoch to the source news. We obtain an accuracy on epoch prediction of almost 78%. This figure indicates that the classificatory works correctly on the source corpus.

We averaged over the classificatory confidence for each epoch separately. We take this parameter as indicator of the similarities between the cognate languages, because, one we know that the classificatory is appropriate, the confidence reflects the similarity. In Table 2 we present the figure for each epoch separately.

As Table 3 shows, the similarity varies over epochs. While these figures are not a direct measure of the similarity of the languages, they represent an objective indication of the high and very high overlapping between the two cognates. In fact was a high pressure for the language spoken in Bessarabia to change, and the Russian influence led to massive changes in the vocabulary, and consequently the similarity dropped significantly. However, the newspaper language preserved much of its identity.

1832-1856	75%	1920-1950	87%
1856-1877	68%	1950-1980	NA
1877-1912	68%	1980-1990	NA
1912-1920	86%	1990-2015	95%

Table 3. Similarity as classifier confidence

3 Conclusions and Further Research

This research presents a diachronic survey conducted to compare journalistic language changes in the Romanian language in terms of time evolution across four regions, Bessarabia, Moldavia, Wallachia and Transylvania. The results highlight major similarities and interesting differences in these collections of publications.

We investigated the problem of diachronic similarity between the mass-media, newspaper, between cognate languages. In particular we focused on the relation between Romanian (including the historical regions: Moldova, Transylvania and Wallachia) and Bessarabian which, started with a high level of similarity and they are again to a very high level of similarity. The method we described is based on statistical analysis of words distributions over epochs reflected on the Romanian printing press and a statistical classifier, SVM, for each epoch. The methodology is language independent and offers an objective quantification of the similarity degree between old Romanian variants.

As further work we plan to expand the methodology farther by including (i) more data, including from period 1945-1990, when in Bessarabia the Latin alphabet was outlawed and (ii) implementing a deeper language analysis using and other statistical classifier as LSTM (Long Short Term Memory) in order to choose the best classifier in diachronic studies. We would like to investigate the semantic similarity between cognates by employing a deep learning approach as well.

Acknowledgments

I would like to thank Dr. Octavian Popescu for his constant guidance, endless suggestions and encouragement and full support to finish this work.

References

[Biber, D., 1987]. D. Biber. *A textual comparison of British and American Writing*. American Speech, (62), pages 99-119, 1987.

[Carstensen, K.-U et al., 2009]. K.-U. Carstensen, C. Ebert, S. Jekat, H. Langer, and R. Klabunde (eds.). *Computerlinguistikund Sprachtechnologie: Eine Einführung*. Spektrum Akademischer Verlag, 2009.

[Cole, R. et al., 1998]. R. Cole, J. Mariani, H. Uszkoreit, G. V. Battista Varile, A. Zaenen, A. Zampolli, V. Zue (eds.) *Survey of the State of the Art in Human Language Technology*. Cambridge University Press, 1998.

[Cristea, D. and Butnariu C., 2004]. D. Cristea, and C. Butnariu. *Hierarchical XML representation for heavily annotated corpora*. In: Proceedings of the LREC 2004 Workshop on XML-Based Richly Annotated Corpora, Lisbon, Portugal, 2004.

[Davies, M., 2013]. M. Davies. *Recent shifts with three non-finite verbal complements in English: Data from the 100-million-word Time corpus (1920s-2000s)*. In: Aarts, Close, Leech and Wallis (eds.) The verb phrase in English: Investigating recent linguistic change with corpora, Cambridge: Cambridge University Press. pages 46-67, 2013.

[Delmestri, A. and Cristianini, N., 2010]. A. Delmestri and N. Cristianini. *String Similarity Measures and PAM-like Matrices for Cognate Identification*. Bucharest Working Papers in Linguistics, 12(2), pages 71-82, 2010.

[Densusianu, O., 1902]. O. Densusianu. *Filologia Romanică în universitatea noastră*, Bucuresci, J. V. Socecu Editeur, page 23, 1902.

[Diaconescu, P., 1974]. P. Diaconescu. *Elemente de istorie a limbii române literare moderne*. Partea I. Probleme de

normare a limbii române literare moderne (1830–1880), Bucureşti, 1974.

[Eggins, S., Martin, J.R., 1997]. S. Eggins, S., J. R. Martin. *Genres and Register of Discourse*. In: Dijk, T.A.v. (ed.) Discourse as Structure and Process (Discourse Studies – A Multidisciplinary Introduction), Vol. 1, pages 231–232. Sage Publications, London, UK, 1997.

[Gîfu, D. et al., 2016]. D. Gîfu, M. Dascălu, Ş. Trăuşan-Matu, and L. Allen.*Time Evolution of Writing Styles in Romanian Language* at the 17th International Conference on Intelligent Text Processing and Computational Linguistics, CICLing 2016, 3-9 Apr. 2016, Konya, Turkey.

[Gîfu, D. and Simionescu, R., 2016]. D. Gîfu, and R. Simionescu. *Tracing Language Variation for Romanian* at the 17th International Conference on Intelligent Text Processing and Computational Linguistics, CICLing 2016, 3-9 Apr. 2016, Konya, Turkey.

[Gîfu, D., 2015]. D. Gîfu. *Contrastive Diachronic Study on Romanian Language*. In: Proceedings FOI-2015, S. Cojocaru, C.Gaindric (eds.), Institute of Mathematics and Computer Science, Academy of Sciences of Moldova, pages 296-310, 2015.

[Gooskens, C., 2006]. C. Gooskens. *Linguistic and extra-linguistic predictors of Inter-Scandinavian intelligibility*. In: Van de Weijer, J. & Los, B. (eds.). Linguistics in the Netherlands, 23, Amsterdam: John Benjamins, pages 101-113, 2006.

[Gooskens, C. et al., 2008]. C. Gooskens, K. Beijering & W. Heeringa. *Phonetic and lexical predictors of intelligibility*. International Journal of Humanities and Arts Computing 2 (1-2), pages 63-81, 2008.

[Jurafsky, D. and Martin, J. H., 2009]. D. Jurafsky and J. H. Martin. *Speech and Language Processing*. Prentice Hall, 2nd edition, 2009.

[Leech, G. at al., 2009]. G. Leech, M. Hundt, C. Mair, and N. Smith. *Change in Contemporary English: A Grammatical Study*. Cambridge: Cambridge University Press, 2009.

[Manning, C. D. and Schütze, H., 1999]. C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

[Popescu, O. and Strapparava, C., 2013]. O. Popescu and C. Strapparava. *Behind the Times: Detecting Epoch Changes using Large Corpora*. In International Joint Conference on Natural Language Processing, Nagoya, Japan, 14-18 October 2013, pages 347-355.

[Popescu, O. and Strapparava, C., 2014]. O. Popescu and C. Strapparava. *Time corpora: Epochs, opinions and changes*. Knowledge-Based Systems, 2014.

[Simionescu, R., 2011]. R. Simionescu. *UAIC Romanian Part of Speech Tagger*, resource on nlptools.info.uaic.ro, "Alexandru Ioan Cuza" University of Iaşi, 2011.

[Trandabăţ D., et al., 2012]. D. Trandabăţ, E. Irimia, V. Barbu Mititelu, D. Cristea, D. Tufiş. *The Romanian Language in the Digital Age*. In: White Paper Series, Georg Rehm and Hans Uszkoreit (eds.), Berlin, Springer, 2012.

[Tufiş, D., Filip, F. Gh., 2002]. D. Tufiş, D., F. Gh. Filip (eds.). *Limba română în Societatea informațională – Societatea Cunoașterii*, Ed. Expert, București, 2002.