

finnsurveytext: Analysis of Open-ended Survey Responses in R

Adeline Clarke , Krista Lagus , Maria Valaste 

Centre for Social Data Science, University of Helsinki

Transformations, A DARIAH Journal Abstract

Volume 1, 2025

<https://transformations.episciences.org>

Dates

Submitted: 15/11/2024

Accepted: 09/04/2025

Published: 24/06/2025

DOI:

[10.46298/transformations.14770](https://doi.org/10.46298/transformations.14770)

© The authors



Creative Commons Attribution 4.0
International

The `finnsurveytext` R package has been created to facilitate the analysis of responses to open-ended survey questions and other structured text data. The package offers a user-friendly, open-source tool and workflow that supports reproducible analysis of text data, including summarisation of response properties, identification of frequent words and phrases, visualisation of responses and creation of a concept network plot. The second version of the package was released in August 2024. It includes integration with the popular `survey` package to allow survey design to be incorporated into the analysis. While the package was created for the analysis of responses written in Finnish, it can also be used to analyse text in other languages. The tool aims to make analysing open-ended questions accessible to social science and humanities scholars without strong programming skills or extensive knowledge of Natural Language Processing methodologies. The objective is to enable the rich data obtained from responses to open-ended questions to be harnessed so that it can be better understood within the context of numeric or categorical data analysis.

Keywords: concept network, `finnsurveytext`, open-ended questions, R, survey design, text analysis

Tiivistelmä

R-paketti `finnsurveytext` on kehitetty helpottamaan kyselytutkimusten avovastauksien ja muiden jäseneltyjen tekstiaineistojen analysointia. Paketti tarjoaa käyttäjäystävällisen, avoimen lähdekoodin työkalun ja työnkulun, joka tukee tekstiaineiston toistettavissa olevaa analyysiä, mukaan lukien vastausten ominaisuuksien tiivistämistä, yleisten sanojen ja fraasien tunnistamista, vastausten visualisointia sekä käsitteverkoston luomista. Paketin toinen versio julkaistiin elokuussa 2024. Se sisältää yhteensopivuuden R-ohjelman suositun `survey`-paketin kanssa, mikä mahdollistaa surveyasetelman sisällyttämisen analyysiin. Vaikka paketti on luotu suomenkielisten vastausten analysointiin, sitä voidaan käyttää myös muiden kielten tekstien analysointiin. Työkalun tavoitteena on tehdä avoimien kysymysten analysointi yhteiskunta- ja humanististen tieteiden tutkijoiden mahdolliseksi ilman syvällisiä ohjelmointitaitoja tai laajoja tietoja luonnollisen kielen käsittelymenetelmistä (NLP). Tavoitteena on mahdollistaa avoimien kysymysten vastauksista saatavan monipuolisen datan hyödyntäminen siten, että sitä voidaan ymmärtää paremmin numeerisen tai kategorisen datan analyysin yhteydessä.

Akateemisen: käsitteverkko, `finnsurveytext`, avovastaukset, R, surveyasetelma, tekstianalyysi

Acknowledgements

The authors would like to thank Joni Oksanen for his contributions in implementing the initial version of the concept network tool, as well as Jaakko Peltonen, Jani-Matti Tirkkonen and Ida-Maria Toivanen for their contributions to the development of `finnsurveytext` through this project, and for their feedback on this manuscript.

Funding

This project has received funding from the European Union–NextGenerationEU instrument and is funded by the Research Council of Finland under grant numbers 345613 and 358721.

Conflict of Interest

The authors do not have any conflict of interest to declare

INTRODUCTION

In the social sciences and humanities, statistical analysis of vast quantities of socially relevant data often relies on ad hoc scripts and processes that are only implemented once, and not necessarily according to best practice using transparent and reproducible analysis techniques. As noted in Turner and Lambert (2015), social scientists may not have the necessary deep understanding of the methods used that would lead to their optimal use. As the use of computational tools increases in the social sciences and humanities, we need to understand that the choices made by and within the tools are not neutral and can significantly impact research to both positive and negative ends (Es, Wieringa, and Schäfer 2018). Thus, the availability of carefully designed, statistically sound and usable analysis tools and workflows remains a crucial avenue for improving the research quality and outcomes of the entire field.

Surveys are a particularly important means for the social scientist or humanities scholar researching humans, their viewpoints and their activities. These may be social surveys, panel surveys or longitudinal surveys. Surveys can be constructed for and queried from population-based samples, allowing generalisations to be made about an entire population. Furthermore, questionnaires may be designed to inquire into the viewpoints or experiences of a focus group appearing in a given context of life, loosely tied to a certain situated social activity or context of life experience, without a connection to population statistics.

Open-ended questions are a useful yet challenging source of survey and questionnaire response data. The use of such questions has a long history and their value has been rediscovered in survey research (Neuert et al. 2021; Singer and Couper 2017). Open-ended questions are particularly useful if researchers do not want to restrict respondents' answers to pre-specified selections. These questions allow respondents to provide diverse answers based on their experience, and often include new answers not previously considered by researchers (He and Schonlau 2021).

However, responses to open-ended questions are often underutilised due to the challenging nature of textual data, especially when survey responses are at a scale no longer feasible for close reading. Social scientists and humanities scholars frequently lack confidence in using the tools and methods of computational text analysis to study large-scale open-ended response data (Turner and Lambert 2015). In addition, the available software is generally “black-box” company-owned tools that do not really benefit researchers, since they obfuscate what was done and why, limiting the scholar's ability to interrogate the data and the analysis process (Schlicht 2020). Few free and open tools exist for analysing open-ended questions, especially for lesser resourced languages such as Finnish, as detailed below. Finnish is a notoriously difficult language to analyse automatically, due to its high level of inflections. For example, a single verb or noun can have more than a thousand inflected forms. Therefore, there is a need for usable and accessible tools to facilitate the analysis of open-ended questions in Finnish.

There are also other sources of structured data that include text fields that could be analysed using tools designed for text data within surveys. For instance, high-volume quantitative data of potential relevance to academic research is generated as a by-product of administrative activities. This includes health records, social services work processes, social benefits applications, or data collected through

educational institutions. Some of these administrative data sources contain not only numeric or class information, but also text fields, either in the words of the employee or written by the recipient of the services. This text data suffers from the same challenges arising from the analysis of responses to open-ended survey questions.

The objective of this article is to present a tool for a previously under-equipped task in the analysis of text that appears in the context of some important quantifying information about a population or a focus group of study. In this work, we not only present such a tool, but also propose an example workflow for analysis. The `finnsurveytext` R package is an open-source resource that addresses the lack of multi-language tools for analysing text data available for social science and humanities scholars. It supports a workflow for data analysis which allows text to be quantified and contextualised, thus helping researchers to harness insights that would otherwise be missed. The name of the tool derives from its initial development for Finnish in particular, although currently the tool can analyse responses in more than 60 languages, as listed in [A](#).

CURRENTLY EXISTING TOOLS FOR FINNISH SURVEYS AND TEXT DATA

Currently, there are insufficient tools for social science and humanities researchers to use in analysing responses to open-ended survey questions. This is largely due to the available tools being costly, difficult to use or only available for English-language text. The currently existing tools are outlined below.

Commercial tools: There are a number of commercial tools available for analysing text data which include support for open-ended survey questions. For instance, ATLAS.ti is a proprietary software typically used by social scientists for qualitative data analysis, such as for analysing interview data (ATLAS.ti 2022). It includes coding, annotation, visualisation and more advanced analysis of text data. The visualisations enabled by ATLAS.ti include word clouds, term frequencies and concept graphs. ATLAS.ti provides machine learning capabilities, including sentiment analysis, Named Entity Recognition (NER) and opinion mining. The tool assumes that text data is unstructured and does not provide specific support for survey analysis. Therefore, to incorporate the findings from text data with other numeric survey data, users must first export these results to a separate platform. Currently, ATLAS.ti can be used to analyse Finnish text, but the software offers no support for dealing with the particular properties of the Finnish language.

LimeSurvey and SurveyMonkey are popular commercial tools for surveys, but they are primarily used for data collection. LimeSurvey does not have built-in functionalities for analysing open-ended survey responses. In contrast, SurveyMonkey does include some paid analysis tools for open-ended questions, such as word clouds, tagging and sentiment analysis, but these have limited proficiency in Finnish. The currently available commercial tools are unable to meet the needs of users in the social sciences and humanities because they are costly, inflexible, limited in scope and provide minimal or no Finnish-language support.

Software packages: In addition to proprietary software, there are a growing number of open-source libraries for analysing text. In R, which is a commonly used programming language in the social sciences, there are a number of packages freely available from CRAN, the language's central software repository. These include those within the `tidyverse` (Wickham et al. 2019) for data manipulation, anal-

ysis and visualisation; `quanteda` (Benoit et al. 2018) for quantitative text analysis; `tidytext` (Silge and Robinson 2016) for text mining; and `koRpus` (Michalke 2020), a text analysis tool that supports multiple languages. Other packages are available for specific text analysis tasks. There is `udpipe` (Wijffels 2023), used for tokenising, tagging and parsing of text data, which supports multiple languages including Finnish; `text2vec` (Selivanov, Bickel, and Wang 2020), used for text vectorisation, topic modelling and word embeddings; `spacyr` (Benoit and Matsuo 2020), which provides an R wrapper for the popular Python Natural Language Processing package `spaCy`; `stm` (Roberts, Stewart, and Tingley 2019), used to create Structured Topic Models from text data; `SentimentAnalysis` (Feuerriegel and Proellocks 2019), used for dictionary-based sentiment analysis; and `textrank` (Wijffels 2020), which applies the TextRank algorithm discussed in section 2.

Additionally, there are many packages available for Python, for example `qualkit` (Wilson 2022), a collection of utilities for conducting qualitative analysis, such as topic modelling, keyword extraction and sentiment analysis of open answers; and the popular Natural Language Toolkit, `NLTK` (Bird, Klein, and Loper 2009), which has significant functionality in Finnish. Moreover, the TurkuNLP group provides a number of extensive Finnish text-analysis tools for use in Python, such as the Turku Neural Parser pipeline (Kanerva et al. 2018), `FinBERT` (Virtanen et al. 2019) and an NER tool (Luoma et al. 2020).

These R and Python packages can be freely used as components in the analysis of open-ended survey responses, but they require users to be comfortable engineering their own text analysis workflows. Additionally, many of these packages provide little or no functionality outside of English-language text, or contain only English-language built-in tools, requiring the user to provide dictionaries for their use in other languages. The exception is the TurkuNLP tools, which instead require a deep understanding of the Natural Language Processing domain. Additionally, these tools do not offer a specific workflow for the analysis of texts in the context of survey data, or other quantitative or categorical variables. As such, the tools currently available in R or Python require a higher level of programming competence than most commercial tools.

OUR CONTRIBUTION: AN OPEN-SOURCE R PACKAGE, *FINNSURVEYTEXT*

`Finnsurveytext` is our response to the need for usable, open-source tools for analysing open-ended survey responses and other text data produced in conjunction with other quantitative or categorical data. This R package was initiated as a work package within *DARIAH-FI*, an infrastructure project for the social sciences and humanities in Finland. It was created to support researchers wanting to conduct transparent exploratory analysis of responses to open-ended survey questions without the use of advanced programming skills or costly tools. While the package was created to support survey analysis of responses written in Finnish, it may be used to analyse text data in many languages and can be usefully employed with any structured text data, such as with administrative data, discussion forums, game logs or social media data.

The package aims to provide a useful and user-friendly set of tools that form a transparent workflow, guiding researchers through the steps of a suitable analysis for the following research purpose: prepare and annotate text data in a suitable standardised format for analysis; conduct common exploratory data-analysis activities that identify trends within the text as well as frequently used terms and

keywords; compare responses (text) between different groups of survey respondents (or sources); and visualise the findings from such analyses. In addition to including familiar plots such as n-gram tables and word clouds, the package facilitates contextualisation of the initial results through our first iteration of a concept network function, as well as functions that facilitate extensive comparison between groups. The package includes comprehensive examples with data designed to help researchers with limited experience in using R or in computational text-analysis methods to understand and use the package's component functions.

The package aims to contextualise text and serve as a bridging tool between qualitative and quantitative analysis. In the second version of the package, we have included a functionality to integrate with the survey package (Lumley 2004) and incorporate survey design, specifically survey weights, within analysis. If desired, the package can be incorporated into more complex workflows by combining its use with other packages and tools for text analysis, such as those highlighted in section 1. Additionally, as the `finnsurveytext` workflow is contained in an R package, it is also easily combined with other survey analysis done in R, enabling the use of a single software for all survey analysis if desired.

INVESTIGATING CONCEPTS THROUGH NETWORKS OF KEYWORDS

It is valuable for a researcher to understand what kind of thoughts or concepts co-occur frequently in the responses to open-ended survey questions or other text data. One way to investigate co-occurring ideas within texts is through a network of keywords, which visualises relative word importance and co-occurrence between words. These networks also attempt to identify when multiple co-occurring words represent the same concept and when some pairs of co-occurring words act as links between different concepts. Producing such a network requires keywords and co-occurring terms to be identified, and their relative importance quantified. Such networks allow for the investigation of common concepts and thoughts within the text. Within such a network, a concept could be identified as a single word, multiple words or the entire plot, depending on the search terms.

Keyword extraction is a well-established research topic that spans a number of research fields, including Natural Language Processing, information retrieval and text mining (Firoozeh et al. 2020). There are a number of methods for the unsupervised extraction of keywords from texts, for example those outlined in Firoozeh et al. (2020) and Nadim, Akopian, and Matamoros (2023). A significant advantage of unsupervised methods is that they do not require training corpora, making them widely adaptable to any language or domain.

One such method is TextRank (Mihalcea and Tarau 2004), a graph-based ranking model, which can extract keywords from open-ended survey responses (or other text). This method determines the importance of a vertex within a graph by taking into account global information computed recursively from the entire graph. The details of the algorithm are outlined in A.

Mihalcea and Tarau (2004) demonstrated that their algorithm performed comparatively well for keyword identification when compared to a contemporary supervised learning approach, which trained an algorithm to determine keywords based on lexical and syntactic features, and had superior performance compared

to determining word importance based on frequency alone. The best results for TextRank were achieved when the co-occurrence window was set to 2 (i.e. edges only created between neighbouring words).

One implementation of the TextRank algorithm in R is the `textrank` package (Wijffels 2020). In this package, a network is created by looking at adjacent words in the text. The weighted PageRank algorithm is used by `textrank` to determine word importance, and edges are created between vertices if the words co-occur, with more co-occurrences increasing edge weight. The default values in the package keep only the top third of words in the text based on their PageRank. The output of this package can then be plotted in R using popular packages such as `ggplot2`.

One example of a concept network being used in research is Medicine Radar (Lagus et al. 2018), an exploratory tool built using existing data from the Suomi24 chat forum containing discussions of health. The tool uses “augmented intelligence,” a combination of automated methods and human inputs, to analyse data in Finnish about medicines and symptoms in order to discover concepts within these forum posts. The Medicine Radar tool is available as an open-source [web interface](#) and can be used to search for medicines and symptoms to find associated concepts as well as actual discussion posts relating to them.

FUNCTIONALITY INCLUDED IN FINNSURVEYTEXT

In this section we describe the `finnsurveytext` package and its component functions. The 39 functions included in the package are outlined in table 1, which is split into data preparation, data exploration, concept network, comparison, and demo app functions. Documentation, including tutorials covering all the functionality, is available on the `finnsurveytext` [website](#).

The data preparation functions can take as input either a dataframe of raw data (containing a column of text data for analysis as well as other—optional—contextual data such as covariates) or a `svydesign` object created using the `survey` package (Lumley 2004). To preprocess raw data for analysis, `finnsurveytext` uses functions from the `udpipe` R package to convert this data into CoNLL-U format (Wijffels 2023). The CoNLL-U format was introduced in 2014 at the Conference of Natural Language Learning. CoNLL-U is a popular annotation scheme often used in Natural Language Processing tasks to tokenise and annotate text. In CoNLL-U format, text is converted to lowercase and split into one line per word. Ten features are then recorded for each word, including an ID, a part-of-speech (POS) tag, the word itself (e.g. “likes”), and the word lemma (e.g. “like”). `Finnsurveytext` currently supports any language model available through `udpipe`, including two Finnish models: the Turku Dependency Treebank (TDT) (Pyysalo et al. 2015) and FinnTreeBank (FTB) (University of Helsinki 2014). The TDT is considered “broad coverage” and includes texts from Wikipedia and news sources, while FTB consists of manually annotated grammatical examples from the Web Version of the Large Grammar of Finnish (VISK). Following annotation of the data into this format, punctuation tokens are removed and stopwords are optionally removed. (Punctuation within a token, such as hyphenated words, is not removed.) Stopword lists are lists of common words (e.g. “and,” “the,” and “is,” or in Finnish, “ja,” “tai,” “olla,” and “yli”), which are often filtered out of the data, leaving less frequently occurring—and thus more meaningful—words remaining. The prepro-

Section	Usage	Functions
Data preparation	Use the <code>udpipe</code> R package to clean and annotate the raw data into a standardised format (CoNLL-U) suitable for analysis.	<code>fst_format()</code> <code>fst_find_stopwords()</code> <code>fst_print_available_models()</code> <code>fst_rm_stop_punct()</code> <code>fst_prepare()</code> <code>fst_format_svydesign()</code> <code>fst_prepare_svydesign()</code>
Data exploration	Create word clouds, n-gram tables and summary tables for initial insights into trends across responses.	<code>fst_summarise_short()</code> <code>fst_summarise()</code> <code>fst_length_summary()</code> <code>fst_pos()</code> <code>fst_use_svydesign()</code> <code>fst_wordcloud()</code> <code>fst_freq_table()</code> <code>fst_ngrams_table()</code> <code>fst_freq_plot()</code> <code>fst_ngrams_plot()</code> <code>fst_freq()</code> <code>fst_ngrams()</code>
Concept network	Creation of a concept network using the <code>textrank</code> R package with node size indicating word importance (PageRank) and edge weight showing co-occurrence of words.	<code>fst_cn_search()</code> <code>fst_cn_edges()</code> <code>fst_cn_nodes()</code> <code>fst_cn_plot()</code> <code>fst_concept_network()</code>
Comparison functions	Corresponding data exploration and concept network functions allowing for comparison between two to four groups of survey respondents.	<code>fst_summarise_compare()</code> <code>fst_length_compare()</code> <code>fst_pos_compare()</code> <code>fst_get_unique_ngrams_separate()</code> <code>fst_get_unique_ngrams()</code> <code>fst_join_unique()</code> <code>fst_ngrams_compare_plot()</code> <code>fst_freq_compare()</code> <code>fst_ngrams_compare()</code> <code>fst_comparison_cloud()</code> <code>fst_cn_get_unique_separate()</code> <code>fst_cn_get_unique()</code> <code>fst_cn_compare_plot()</code> <code>fst_concept_network_compare()</code>
Demo app	An RShiny package demo.	<code>runDemo()</code>

Table 1: Overview of functions in the `finnsurveytext` R package.

cessing functions also allow inclusion of other data (in addition to the CoNLL-U columns), such as a weight column or covariate columns, which can be used to split the data into cohort or treatment-control groups in the comparison functions.

To support exploratory analysis of open-ended questions, `finnsurveytext` includes functions which summarise the responses in terms of length and type of words, identify and plot the most frequent words and n-grams, and produce word clouds from responses. An n-gram is a contiguous sequence of n words, where n is an integer. For instance, a 3-gram, which is also known as a trigram, is a set of three words in order. Within `finnsurveytext`, n-grams are created after pre-processing raw data, including lemmatisation and the removal of stopwords and punctuation (including sentence boundaries). For example, if the trigram “menee yli hilseen” existed in the raw data, it would be found as the bigram “mennä hilse” in processed data. The results of these functions can help researchers better understand their data, create hypotheses based on these initial insights and inform future analysis. These functions can be weighted (using the `weights` column or provision of a relevant `svydesign` object) or normalised by number of words or responses.

The `finnsurveytext` package currently contains our first iteration of a function that plots a concept network. These plots visualise keywords, which are identified through the TextRank algorithm, and maps co-occurrences between these terms. Vertices represent words, with vertex size indicating word importance, while co-occurrence between words is shown through edges, with edge thickness indicating number of co-occurrences. Word importance is determined recursively (through the unsupervised TextRank algorithm), with words getting more weight based on how many words co-occur and the weight of these co-occurring words. The concept network functions take search terms input by the user and the algorithm, then suggest other words that are related to these input terms by co-occurrence. The input terms can be identified through functions in the package (such as `fst_cn_search()` or `fst_freq_table()`) or through other analysis conducted separately by the user. The concept network functions can be used to identify concepts, which could be individual words or a group of co-occurring words, or may contain a single “concept” whose component words are investigated and identified within a single network plot.

To enable determination of whether different groups of survey participants have, in general, responded differently to the open-ended question (or authors have produced different types of text based on contextual factors), `finnsurveytext` includes a counterpart comparison function for each analysis function, which is intended to be used to compare responses between groups. One way to split the data is by using a different question within a survey, such as a categorical question (e.g. gender, location or level of education) or an ordinal variable (such as age or income bracket). These counterpart functions highlight differences that are identified between each group. The comparison functions can be used to arbitrarily compare many groups of respondents.

Finally, a beta RShiny demo app has been created as a prototype of a user interface that could be used to access the `finnsurveytext` functionality without requiring the user to write code.

EXAMPLE WORKFLOW

In this section we will demonstrate the use of `finnsurveytext` to analyse responses to an open-ended survey question. Since the package is intended to be used in conjunction with other analyses, potentially as a data exploration and hypothesis identification tool, this workflow should be considered a demonstration of the package rather than a complete investigation of our example survey question. In a real research setting, considerable analysis would likely remain, informed by the initial findings facilitated by our package. The workflow is split into five sections: data preparation, data exploration, creation of a single concept network plot, comparison of different respondent groups, and creation of a comparison concept network plot. Following the workflow, we demonstrate how the package can be used for non-Finnish text and show the beta RShiny app that we have built for the package.

DATA PREPARATION

The survey contains response data from 945 participants. As revealed through the summary functions described below, this question was answered by 920 respondents (97% response rate), of which there were 660 females (98% response rate), 175 males (96% response rate) and 85 respondents who did not provide a gender (96% response rate).

Following package installation, we use the `fst_prepare()` function to format the data into the CoNLL-U format and remove punctuation and stopwords from the results. The remaining analysis functions require data in this format. We have formatted the data using the FinnTreeBank language model, removed stopwords from the “nltk” list, added a column for responses to the “gender” question used to demonstrate the comparison functions, and included survey weights from the “paino” column in the raw data. The first time you run `fst_prepare()` with a particular language model it will be downloaded and saved into your working directory.

```
R> library(finnsurveytext)
R> prep_df <- fst_prepare(dev_coop,
+                         question = "q11_3",
+                         id = "fsd_id",
+                         stopword_list = "nltk",
+                         model = "ftb",
+                         weights = "paino",
+                         add_cols = "gender"
+ )
```

The top six rows of the formatted data are shown in table 3 within [A](#).

Alternatively, we can format the data from a `svydesign` object to produce the same result. For demonstration, we create a `svydesign` object called `svy_dev_coop` as below:

```

R> svy_dev_coop <- survey::svydesign(id = ~1,
+                               weights = ~paino,
+                               data = dev_coop
+ )
R> prepd_df_2 <- fst_prepare_svydesign(svydesign = svy_dev_coop,
+                                   question = "q11_3",
+                                   id = "fsd_id",
+                                   model = "ftb",
+                                   use_weights = TRUE,
+                                   add_cols = "gender"
+ )

```

DATA EXPLORATION

Now that the formatted data is stored in `prepd_df`, we can explore the data to find initial insights. As our starting exploratory analysis, we produce three summary tables. First, `fst_summarise()` creates a summary table for the input CoNLL-U data. This provides the response count and proportion, total number of words, the number of unique words and the number of unique lemmas. This table shows that of 945 survey participants, 920 responded to this open-ended question (97% of participants).

```
R> fst_summarise(prepd_df, desc = "All")
```

	Description	Respondents	No Response	Proportion
1	All	920	25	0.97
	Total Words	Unique Words	Unique Lemmas	
	4192	1132	994	

Next, `fst_length_summary()` summarises the distribution of response lengths. Since we have removed stopwords and punctuation from the data, these are not included in the summaries. As such, the response lengths and word-type distributions shown in these tables differ from those in the raw data. This table tells us that most responses consist of a single sentence of a few words.

```
R> fst_length_summary(prepd_df, desc = "All")
```

	Description	Respondents	Mean	Minimum	Q1	Median	Q3	Maximum
1	All- Words	920	5.52	1	4	5	6	32
2	All- Sentences	920	1.01	1	1	1	1	3

Finally, `fst_pos()` creates a summary table that counts the number of words for each POS tag within the preprocessed data. From this we can see that the responses are mostly nouns (79% of words), verbs (7% of words) and adjectives (1% of words).

```
R> fst_pos(prepd_df)
```

	UPOS	UPOS_Name	Count	Proportion
1	ADJ	adjective	389	0.093
2	ADP	adposition	24	0.006
3	ADV	adverb	64	0.015
4	AUX	auxiliary	3	0.001
5	CCONJ	coordinating conjunction	3	0.001
6	DET	determiner	28	0.007
7	INTJ	interjection	2	0.000
8	NOUN	noun	3286	0.789
9	NUM	numeral	5	0.001
10	PART	particle	29	0.007
11	PRON	pronoun	12	0.003
12	PROPN	proper noun	31	0.007
13	PUNCT	punctuation	0	0.000
14	SCONJ	subordinating conjunction	0	0.000
15	SYM	symbol	1	0.000
16	VERB	verb	278	0.067
17	X	other	12	0.003

The formatted data can also be used to produce visualisations of the responses. For instance, figure 1 shows a word cloud of all word types (all POS tags), produced using the `fst_wordcloud()` function, while figure 2 shows the word cloud produced when weights are included. Again, only remaining words (no stopwords) are included in these plots.

```
R> fst_wordcloud(prepd_df)
```



Figure 1: Word cloud.

13

Figure 2: Weighted word cloud.

The most frequent words and n-grams are found using the `fst_freq()` and `fst_ngrams()` functions. In figures 3 and 4 we show the 10 most common words and bigrams (sets of two words, in order) weighted by the response weights included in the data. Common multiword concepts can be identified through n-grams tables, such as the bigrams table below (figure 4), which identified “puhdas vesi” (clean water) and “vesi puute” (water shortage) as two common two-word phrases. (In fact, if trigrams are plotted, we find that “puhdas vesi puute” [shortage of clean water] is the most common three-word phrase in the responses.) Order matters in n-grams, so the bigram “puhdas vesi” is considered different to “vesi puhdas” despite containing the same words.

```
R> fst_freq(prepd_df, use_column_weights = TRUE)
```

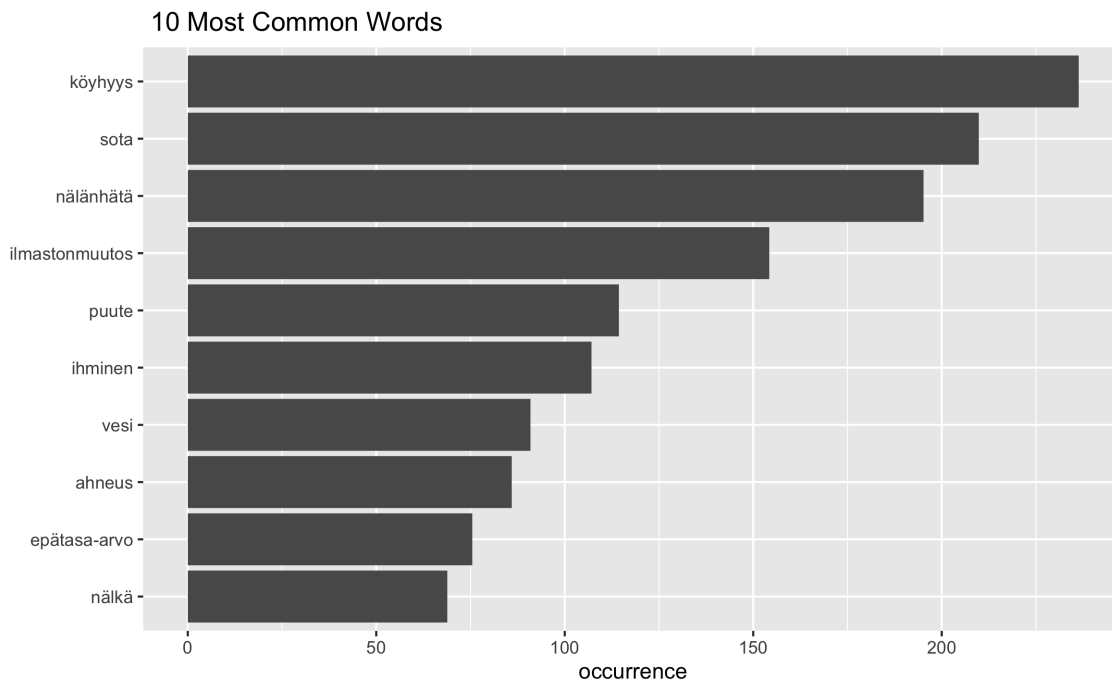


Figure 3: Most frequent words.

```
R> fst_ngrams(prepd_df, ngrams = 2, use_column_weights = TRUE)
```

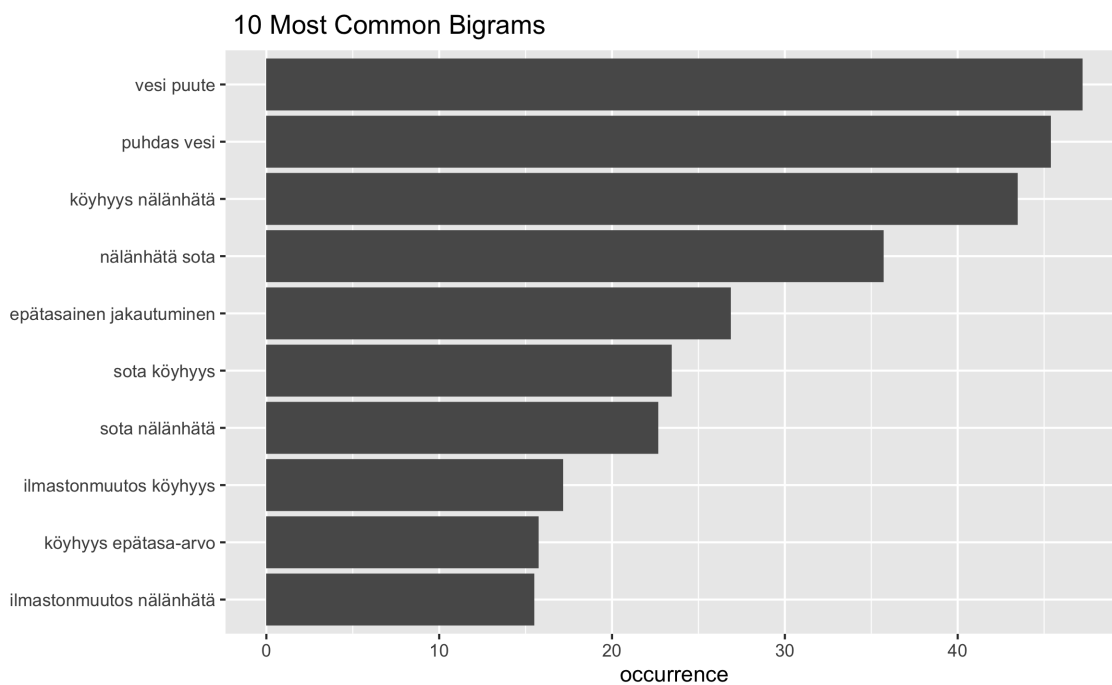


Figure 4: Most frequent bigrams.

CREATION OF A CONCEPT NETWORK PLOT

Figure 5 contains a concept network plot based on the three most frequent words in the responses: “köyhyys” (poverty), “nälänhätä” (famine) and “sota” (war). The concept words input by the user are shown in red in figure 5. These concept networks can be customised using the concept network functions in table 1 or via the single function `fst_concept_network()`. The plot shows that “vesi” (water) is mentioned in the context of “vesi puute” (water shortage) and “puhdas vesi” (clean water), indicating that lack of clean water is a common theme found in the texts. This could be investigated in more detail through further use of the `fst_concept_network()` function.

```
R> fst_concept_network(prepd_df,
+                       concepts = "köyhyys, nälänhätä, sota",
+                       title = "Concept Network",
+                       norm = "number_resp"
+ )
```

Concept Network

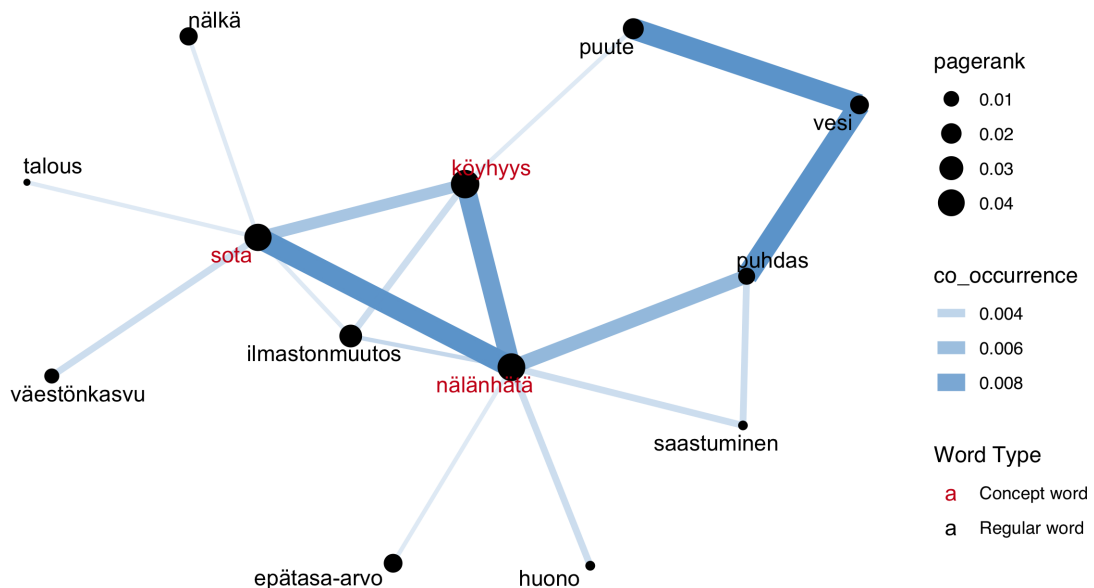


Figure 5: Concept network.

COMPARISON OF RESPONDENT GROUPS

To demonstrate our functions and compare responses to the open-ended questions between groups, we will partition responses by the “gender” background question in the data. We will include responses where “gender” has not been provided (survey participants had only two options for gender, indicating “male” or “female” respectively), as this may represent a marginalised group within the respondents, but this may not be advisable if there are too few respondents in this group.

The first set of comparison functions are `fst_summarise_compare()`, `fst_length_compare()` and `fst_pos_compare()`. These functions are similar to `fst_summarise()`, `fst_length()` and `fst_pos()` but create tables split by group. From these results, users can identify that while most responses are from female survey participants, all genders have a high response rate for this question, use similar types of words and provide a similar-length response.

```
R> fst_summarise_compare(prepd_df,
+                         field = "gender",
+                         exclude_nulls = FALSE,
+                         rename_nulls = "Gender NA"
+ )
```

	Description	Respondents	No Response	Proportion	Total Words
1:	Female	660	13	0.98	2993
2:	Gender NA	85	4	0.96	404
3:	Male	175	8	0.96	795
	Unique Words	Unique Lemmas			
1:	823	722			
2:	225	208			
3:	383	354			

```
R> fst_length_compare(prepd_df,
+                     field = "gender",
+                     incl_sentences = TRUE,
+                     exclude_nulls = FALSE,
+                     rename_nulls = "Gender NA"
+ )
```

	Description	Respondents	Mean	Minimum	Q1	Median	Q3	Maximum
1:	Female- Words	660	5.518	1	4	5	6	28
2:	Female- Sentences	660	1.012	1	1	1	1	3
3:	Gender NA- Words	85	5.694	3	4	4	6	18
4:	Gender NA- Sentences	85	1.012	1	1	1	1	2
5:	Male- Words	175	5.417	2	4	5	6	32
6:	Male- Sentences	175	1.029	1	1	1	1	3

```
R> fst_pos_compare(prepd_df,
+                  field = "gender",
+                  exclude_nulls = FALSE,
+                  rename_nulls = "Gender NA"
+ )
```

	UPOS	Part_of_Speech_Name	Female-Count	Female-Prop
1	ADJ	adjective	276	0.418
2	ADP	adposition	19	0.029
3	ADV	adverb	44	0.067
4	AUX	auxiliary	0	0.000
5	CCONJ	coordinating conjunction	2	0.003
6	DET	determiner	24	0.036
7	INTJ	interjection	1	0.002

8	NOUN	noun	2373	3.595
9	NUM	numeral	1	0.002
10	PART	particle	18	0.027
11	PRON	pronoun	8	0.012
12	PROPN	proper noun	19	0.029
13	PUNCT	punctuation	0	0.000
14	SCONJ	subordinating conjunction	0	0.000
15	SYM	symbol	1	0.002
16	VERB	verb	195	0.295
17	X	other	12	0.018

	Gender NA-Count	Gender NA-Prop	Male-Count	Male-Prop
1	44	0.518	69	0.394
2	2	0.024	3	0.017
3	9	0.106	11	0.063
4	0	0.000	3	0.017
5	1	0.012	0	0.000
6	0	0.000	4	0.023
7	1	0.012	0	0.000
8	302	3.553	636	3.634
9	3	0.035	1	0.006
10	3	0.035	8	0.046
11	1	0.012	3	0.017
12	0	0.000	12	0.069
13	0	0.000	0	0.000
14	0	0.000	0	0.000
15	0	0.000	0	0.000
16	38	0.447	45	0.257
17	0	0.000	0	0.000

The comparison plot functions can arbitrarily plot many groups in a grid for comparison between multiple groups. Words featuring in the top 10 words for only one group of respondents are coloured for emphasis. Figure 6 shows a comparison plot of the most frequent words. We have excluded responses where no gender was provided, simply to make the plot easier to view in this format.

```
R> fst_freq_compare(prepd_df,
+                   "gender",
+                   use_column_weights = TRUE,
+                   exclude_nulls = TRUE,
+                   rename_nulls = "Gender NA"
+ )
```

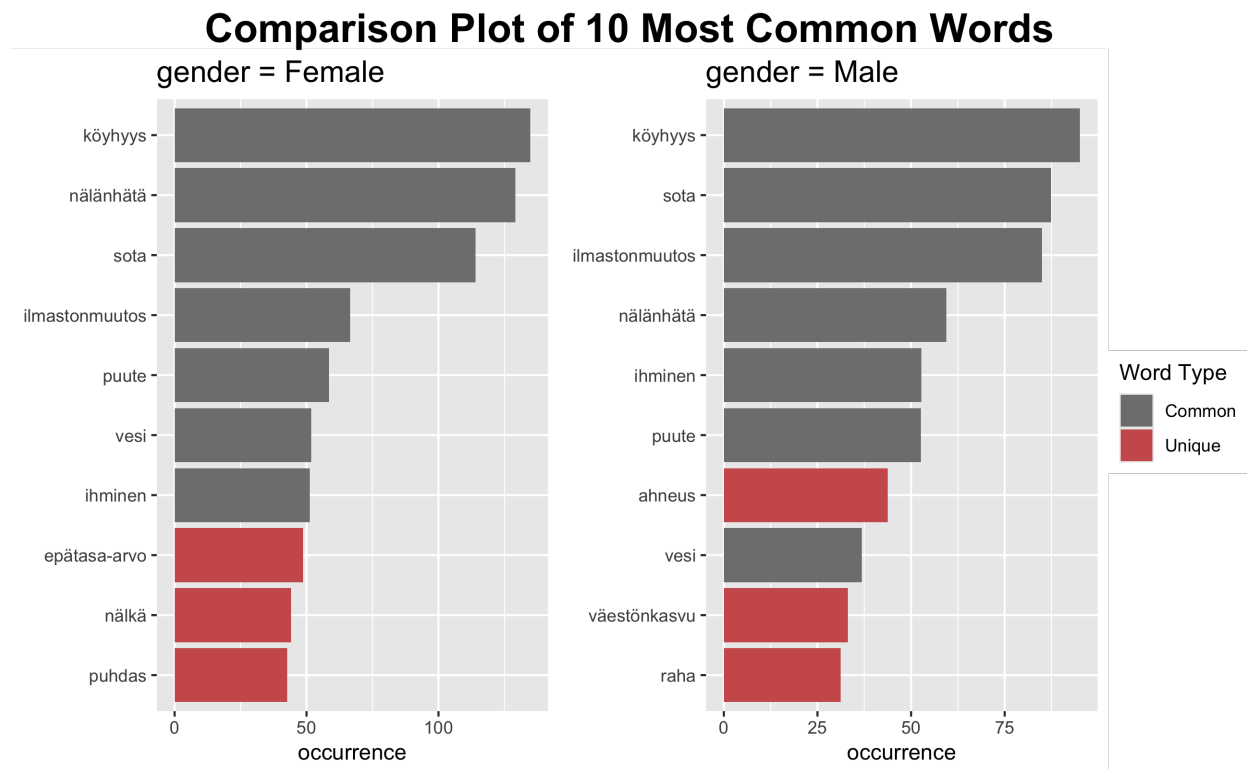


Figure 6: Comparison of most frequent words in female and male responses.

CREATION OF A COMPARISON CONCEPT NETWORK PLOT

For the comparison concept network in figure 7, an additional term, “ilmastonmuutos” (climate change), which was not included in the single concept network, has been added, as this was identified through the comparison plot of most frequent words (figure 6) as a term more frequent in male responses than female. From these concept networks we can observe that use of our four searched terms—“ilmastonmuutos,” “nälänhätä,” “köyhyys” and “sota” (climate change, famine, poverty, war)—differs overall by gender. In particular, the responses of females (n=660) emphasise a strong connection between the central concepts of war and famine, whereas many other terms appear connected to either one or the other. The figure provides a rich and nuanced view of the conceptual relations within the female responses. In contrast, the male responses (N=175) show only the four key concepts, where poverty is central and all three others form a strong connection to that one. No other keywords even appear in the figure. Figure 7 shows the concept network plot for responses by female and male survey participants created using `fst_freq_compare()` and `fst_concept_network_compare()`.

```
R> fst_concept_network_compare(prepd_df,
+                             "gender",
+                             concepts = "köyhyys, nälänhätä, sota,
+                                       ilmastonmuutos" ,
+                             exclude_nulls = TRUE,
+                             rename_nulls = "Gender NA"
+ )
```

The fact that the concept network for males shows less detail and fewer terms than the concept network for females might in a general case be due to one of

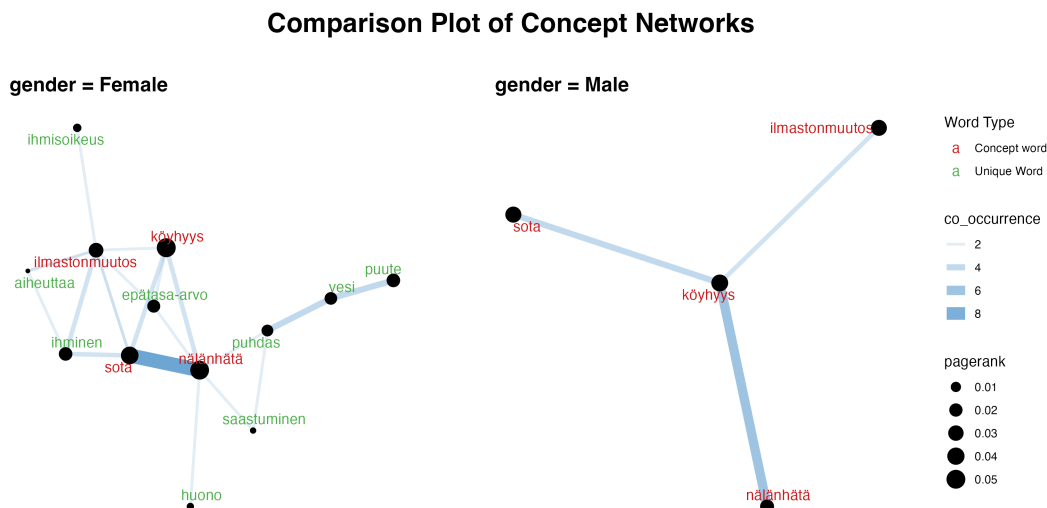


Figure 7: Comparison concept network of female and male responses.

the following: (1) there might be considerably fewer male responses than female responses, (2) the male responses might be significantly shorter than the female responses, or (3) the male responses might in general exhibit fewer notable patterns of connections, concentrating on only one theme at a time, whereas the female respondents might have more commonly been considering the problem from many different angles, which then emerge in the richness of the resulting concept network. We know from previous analyses that there were 660 responses to this question from female survey participants, compared to 175 responses from males. This means that there was over three times more data from female responses, supporting our first theory (1). A plausible reason for the lack of detail in the concept graph for males may therefore simply be the fewer number of responses from males. We also know from our summary tables that the second theory (2) is not correct, since there was a similar response length between the genders. To explore our third theory would require additional analysis into the responses and could be a subject for further exploration. It is noteworthy that when the number of responses is very small, a concept network does not even appear, because co-occurrences of key terminology within the responses are not frequent enough.

USING FINNSURVEYTEXT IN OTHER LANGUAGES

To demonstrate functionality in languages other than Finnish, a sample survey in English is included with the package. In this section we will demonstrate how to format this dataset using functions from the “Data preparation” section in table 1. Once the data is formatted, all functions and analyses work identically regardless of the language used in the text.

The example survey with English text is included in the package as `english_sample_survey`. It contains answers to the question, “Joe’s doctor told him that he would need to return in two weeks to find out whether or not his condition had improved. But when Joe asked the receptionist for an appointment, he was told that it would be over a month before the next available appointment. What should Joe do?” This data is available upon registration from

the GESIS Data Archive for the Social Sciences under a CC BY 4.0 licence (Schonlau 2022). The top six rows of raw data are shown in table 4 within A.

Before formatting the data, we need to determine which treebank we will use for formatting the data and which stopwords list to use. We can find all the English stopwords lists available for use within our package by running `fst_find_stopwords()` with `language = "en"`, since "en" is the ISO code for English.

```
R> fst_find_stopwords(language = "en")

# A tibble: 5 × 3
  Name      Stopwords      Length
  <chr>    <list>      <list>
1 marimo  <chr [237]> <int [1]>
2 nltk    <chr [179]> <int [1]>
3 smart   <chr [571]> <int [1]>
4 snowball <chr [175]> <int [1]>
5 stopwords-iso <chr [1,298]> <int [1]>
```

There is also a function to print all the treebanks available for use with a specific language in `finnsurveytext`, which saves users having to find these in the `udpipe` documentation.

```
R> fst_print_available_models(search = "english")
[1] "english-ewt"      "english-gum"      "english-lines"    "english-partut"
```

Thus, we can choose "english-ewt" as our model and "nltk" as our stopwords list to prepare our English text data for further analysis. The top six rows of the formatted data are shown in table 5 within A.

```
R> prepd_en <- fst_prepare(english_sample_survey,
+                           question = "text",
+                           id = "id",
+                           model = "english-ewt",
+                           stopwords_list = "nltk",
+                           language = "en")
```

DEMO APP

Currently, the RShiny demo app is only available via R. It is accessed by running `runDemo()` and can be used to produce all analysis of Finnish text enabled through the package via a user interface. Future development will expand the user interface to include all other languages available to `finnsurveytext` and to host the app externally. Some screenshots of the beta app are included below, in figures 8, 9 and 10.

```
R> runDemo()
```

'finnsurveytext' package demo BETA

Instructions Prepare Data Explore Data Compare Groups of Responses

Load Data

Load Data

Format Data

Format Data

Instructions

finnsurveytext functions require data formatted into CoNLL-U format. To learn more about the format, see the [Universal Dependencies Project](#).

This panel is used to format the data for later steps.

Use the dropdowns to choose which columns in your data contain your open-ended question, the IDs, and whether to include weights or other additional columns. Pick which Finnish language model to use and which list of stopwords to remove from the data.

When you're ready to format the data, click the button below.

(The only way to weight your data in this demonstration is via the "weights from column" since the use of *sydesign* objects is not demonstrated in this app.)

Format your data

Which question/column contains the open-ended question?

fsd_id

Which question/column contains the id?

fsd_id

Which Finnish language model should we use?

☒ fts

☐ tdt

Which stopwords list should we use?

☒ rtk

☐ snowball

☐ stopwords-iso

☐ none

OPTIONAL: Which question/column contains the weights?

NO WEIGHTS

OPTIONAL: Which additional columns should we include in the formatted data?

Press this to format your data Press this to toggle whether to show your formatted data

Figure 8: Within the “Prepare Data” tab of the demo app, users can upload and format their data for analysis with *f_{innsurvey}text*.

'finnsurveytext' package demo BETA

Instructions Prepare Data Explore Data Compare Groups of Responses

Summary Tables

Summary Tables

Wordcloud

Wordcloud

Frequent Words/Phrases

N-grams

Concept Network

Concept Network

N-grams

A n-gram is a set of N words in order.

The tab is used to create a plot of the most common words/phrases in your formatted data.

Use the dropdowns to indicate what size n-gram you want to plot and how many n-grams to show. You can also indicate if you want to normalise the data and/or use weights and exclude word types if you want to. Also, you can indicate whether to strictly cut-off at the cut-off number or show equally-occurring words.

What size n-gram should we show? (To show top words, choose 1)

1

How should we deal with ties?

☒ strict cut-off, show first-occurring

☐ alphabetically

☐ show ties

How many words/phrases should we show?

10

Should we normalise the data?

☒ NULL (pick this also if you want to use weights)

☐ number of words

☐ number of responses

Do you want to weight responses in table?

☒ no weights

☐ weights from formatted data

Would you like to add a name to plot title?

Press this to make (or refresh) the n-gram plot

Untick any word types you want to exclude from the plot.

☒ ADJ

☒ ADP

☒ ADV

☒ AUX

☒ CCONJ

☒ DET

☒ INTJ

☒ NOUN

☒ NUM

☒ PART

☒ PRON

☒ PROPN

☒ PUNCT

☒ SCONJ

☒ SYM

☒ VERB

☒ X

Figure 9: The “Explore Data” tab of the demo app includes functionality to identify frequently occurring n-grams in the data, as well as other exploratory functions.

'finnsurveytext' package demo BETA

Instructions Prepare Data Explore Data Compare Groups of Responses

Comparison Functions

Which field would you like to use to split the data for comparison?

Would you like to exclude nulls in the comparison field?

☒ Yes

☐ No

There are counterpart comparison functions for each of the functions in the previous "Explore Data" tab.

Recall that when you preprocessed the data, you were given the option to include additional columns. These columns can now be used to allow for comparison between respondents based on these values.

On the left, you can pick which column to use to split the data, and also indicate what to do with responses which have a null in this splitting column.

Comparison Summary Tables

As previously, you can pick which summary table to show here.

Which comparison summary table would you like to see?

☒ response

☐ length

☐ part-of-speech

Press this to make (or refresh) the table

Comp. Tables

Comparison Summary Tables

Comp. Cloud

Comparison Cloud

Comp. of Freq. Words

Comparison N-grams

Comp. Concept Network

Comparison Concept Network

Figure 10: The “Compare Groups of Responses” tab of the demo app can be used to identify key differences in responses based on covariate data.

DISCUSSION AND NEXT STEPS

Future work is planned to expand the package. This will extend the workflow and include other tools and methodologies for the analysis of text data. One possible direction for this work could include the use of a background language model

or word embeddings to calculate distances for the concept network calculation. We will also include additional unsupervised methods for keyword extraction, such as those compared in Nadim, Akopian, and Matamoros (2023) as alternatives to the TextRank algorithm. We may also expand the `finnsurveytext` workflow to enable integration of a social theory perspective into network calculations, such as by importing a dictionary. This would help researchers wishing to operationalise a particular social theory and analyse it from text data in a transparent and reproducible manner.

Another area of further improvement will be the continued development of the graphical user interface for non-programmers and users unfamiliar with text analytics. This is expected to facilitate end-user adoption and considerably widen the user base within the social sciences and humanities. Furthermore, we will implement an interactive browsing and exploration interface that starts from the obtained concept network and allows the user to click on a concept to obtain access to the responses where that concept exists, with the concept highlighted in the responses (much as was implemented in the Medicine Radar concept network interface) (Lagus et al. 2018). This will facilitate explorative and qualitative research on the responses. Functionality will also be included so that users can click on a concept and see other measures and characteristics of respondents who utilised that concept in their responses.

In preparation for this development of `finnsurveytext`, we will engage with literature and best practice coming from tool criticism research, such as Es (2023) and Es, Wieringa, and Schäfer (2018), and seek feedback from potential users through surveys, interviews and code walkthroughs to understand how the tool is used and, critically, how it influences the research resulting from its use. In doing so, we will better understand our users, improve usability of the tool, clarify any misunderstandings evident in its use, and take steps to improve the transparency of the tool. This feedback will either strengthen our claim that `finnsurveytext` meets the needs of social science and humanities scholars in the analysis of responses to open-ended survey questions and other text data, or help us better understand how to adapt and expand the tool to meet these needs.

In the analysis section we noted that when the number of answers drops, so the detail from the concept network diminishes, and eventually no concept network is produced at all. To enable analysis of very small data, such as tens of responses rather than hundreds, and when the answers are relatively short text segments, we will incorporate additional text-analysis methods into the package. These additions could require the application of Large Language Models, vector embeddings or transformer models calculated based on very large auxiliary datasets. Such models are needed to tease out relationships between latent properties that are not directly expressed in the texts themselves. However, since adding such latent analysis introduces many issues to consider regarding the suitability of these external language models, we have left it as a potential avenue for future development.

Although our demonstration of `finnsurveytext` has shown how the tool can be used in a research workflow for survey and questionnaire analysis, the nature of the tool points directly to its potential relevance in context with many administrative data sources that include text fields. Exploring the usefulness of this tool and this type of analysis workflow for administrative data sources, such as from health services, education administration, social work, the allocation of social benefits,

tax records or within democratic processes and citizen participation, remain potential and possible areas for future empirical work.

CONCLUSION

The responses to open textual questions contain rich but underutilised data within surveys. These questions can also provide important context for other results within the survey. Researchers may additionally want to know how different groups of respondents differ in their responses to open-ended questions. Open-ended questions, including those not answered in English, can be analysed using `finnsurveytext` in a statistically sound and transparent manner.

The `finnsurveytext` R package and workflow has been created to enable the analysis of responses to open-ended survey questions and other structured text data by researchers who may not otherwise be able to access or use existing tools or methodologies. The `finnsurveytext` package contains a suite of tools available for the analysis of text data, including language-specific annotation, to enable both Finnish and a range of other languages to be used with the tool. The package enables users to format, explore and visualise text, and to incorporate these findings with other analyses, such as those from other questions within a survey. Analysis and visualisations from surveys can be weighted according to survey design through integration with the `survey` package, or through weights included in the raw data. The package is intended to be useful and user-friendly so that text data from open-ended survey questions can be examined and understood, rather than remaining underutilised in research.

COMPUTATIONAL DETAILS

The results in this paper were obtained using R 4.3.0 with the `finnsurveytext` 2.1.1 package. R itself and all packages used are available from the [Comprehensive R Archive Network \(CRAN\)](#).

The `finnsurveytext` package has the following dependencies: `data.table`, `dplyr`, `ggplot2`, `ggpubr`, `ggraph`, `igraph`, `magrittr`, `purrr`, `RColorBrewer`, `stopwords`, `stringr`, `textrank`, `tibble`, `tidyr`, `udpipe`, `wordcloud`.

The development version of the package is also available directly from [GitHub](#). Documentation, including tutorials covering all the functionality, is available on the `finnsurveytext` [website](#). The replication script for the example workflow in section 4 can be found on [GitHub](#).

REFERENCES

- ATLAS.ti. 2022. *ATLAS.ti Windows: User Manual*. ATLAS.ti. <https://atlasti.com/manuals-and-documents>.
- Benoit, Kenneth, and Akitaka Matsuo. 2020. *spacyr: Wrapper to the “spaCy” “NLP” Library*. R package version 1.2.1. The CRAN Comprehensive R Archive Network. <https://CRAN.R-project.org/package=spacyr>.
- Benoit, Kenneth, Kohei Watanabe, Haiyan Wang, Paul Nulty, Adam Obeng, Stefan Müller, and Akitaka Matsuo. 2018. “Quanteda: An R Package for the Quantitative Analysis of Textual Data.” *Journal of Open Source Software* 3 (30): 774. <https://doi.org/10.21105/joss.00774>.
- Bird, Steven, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O’Reilly Media, Inc.
- Brin, Sergey, and Lawrence Page. 1998. “The Anatomy of a Large-scale Hypertextual Web Search Engine.” *Computer Networks and ISDN Systems* 30 (1): 107–117. [https://doi.org/10.1016/S0169-7552\(98\)00110-X](https://doi.org/10.1016/S0169-7552(98)00110-X).
- Es, Karin van. 2023. “Unpacking Tool Criticism as Practice, in Practice.” *Digital Humanities Quarterly* 17 (2). <https://www.digitalhumanities.org/dhq/vol/17/2/000692/000692.html>.
- Es, Karin van, Maranke Wieringa, and Mirko Tobias Schäfer. 2018. “Tool Criticism: From Digital Methods to Digital Methodology.” In *WS.2 2018 Proceedings of the 2nd International Conference on Web Studies*, 24–27. New York: Association for Computing Machinery. <https://doi.org/10.1145/3240431.3240436>.
- Feuerriegel, Stefan, and Nicolas Proelochs. 2019. *SentimentAnalysis: Dictionary-Based Sentiment Analysis*. R package version 1.3-3. The CRAN Comprehensive R Archive Network. <https://CRAN.R-project.org/package=SentimentAnalysis>.
- Finnish Children and Youth Foundation. 2019. “Young People’s Views on Development Cooperation 2012.” Dataset. Version 2.0. <http://urn.fi/urn:nbn:fi:fsd:T-FSD2821>.
- Firoozeh, Nazanin, Adeline Nazarenko, Fabrice Alizon, and Béatrice Daille. 2020. “Keyword Extraction: Issues and Methods.” *Natural Language Engineering* 26, no. 3 (May): 259–291. <https://doi.org/10.1017/S1351324919000457>.
- He, Zhoushanyue, and Matthias Schonlau. 2021. “Coding Text Answers to Open-ended Questions: Human Coders and Statistical Learning Algorithms Make Similar Mistakes.” *Methods, data, analyses (mda)* 15, no. 1 (January): 17. <https://doi.org/10.12758/mda.2020.10>.
- Kanerva, Jenna, Filip Ginter, Niko Miekka, Akseli Leino, and Tapio Salakoski. 2018. “Turku Neural Parser Pipeline: An End-to-End System for the CoNLL 2018 Shared Task.” In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies (Brussels, Belgium, October)*, 133–142. Kerrville, TX: Association for Computational Linguistics. <https://doi.org/10.18653/v1/K18-2013>.
- Lagus, Krista, Minna Ruckenstein, Atte Juvonen, and Chang Rajani. 2018. “Medicine Radar—A Tool for Exploring Online Health Discussions.” In *Proceedings of the Digital Humanities in the Nordic Countries 3rd Conference (Helsinki, Finland, 7–9 March)*, 460–468. CEUR-WS.org. <http://ceur-ws.org/Vol-2084/>.
- Lumley, Thomas. 2004. “Analysis of Complex Survey Samples.” *Journal of Statistical Software* 9 (8): 1–19. <https://doi.org/10.18637/jss.v009.i08>.
- Luoma, Jouni, Miika Oinonen, Maria Pyykönen, Veronika Laippala, and Sampo Pyysalo. 2020. “A Broad-coverage Corpus for Finnish Named Entity Recognition.” In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, edited by Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, et al., 4615–4624. Marseille, France: European Language Resources Association. <https://aclanthology.org/2020.lrec-1.567/>.
- Michalke, Meik. 2020. *koRpus: An R Package for Text Analysis*. Version 0.13-3. <https://reaktanz.de/?c=hacking&s=koRpus>.

- Mihalcea, Rada, and Paul Tarau. 2004. "TextRank: Bringing Order into Text." In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (Barcelona, Spain, 25–26 July)*, edited by Dekang Lin and Dekai Wu, 404–411. Kerrville, TX: Association for Computational Linguistics. <https://aclanthology.org/W04-3252>.
- Nadim, Mohammad, David Akopian, and Adolfo Matamoros. 2023. "A Comparative Assessment of Unsupervised Keyword Extraction Tools." *IEEE Access* 11:144778–144798. <https://doi.org/10.1109/ACCESS.2023.3344032>.
- Neuert, Cornelia E., Katharina Meitinger, Dorothee Behr, and Matthias Schonlau. 2021. "Editorial: The Use of Open-ended Questions in Surveys." *Methods, data, analyses (mda)* 15 (1): 3–6. <https://mda.gesis.org/index.php/mda/article/view/366>.
- Pyysalo, Sampo, Jenna Kanerva, Anna Mäsilä, Veronika Laippala, and Filip Ginter. 2015. "Universal Dependencies for Finnish." In *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015) (Vilnius, Lithuania, 11–13 May)*, edited by Beáta Megyesi, 163–172. Linköping: Linköping University Electronic Press. <https://aclanthology.org/W15-1821>.
- Roberts, Margaret E., Brandon M. Stewart, and Dustin Tingley. 2019. "stm: An R Package for Structural Topic Models." *Journal of Statistical Software* 91 (2): 1–40. <https://doi.org/10.18637/jss.v091.i02>.
- Schlicht, Helene. 2020. "Open Access, Open Data, Open Software? Proprietary Tools and Their Restrictions." In *Digital Methods in the Humanities: Challenges, Ideas, Perspectives*, edited by Silke Schwandt, 25–58. Bielefeld: Bielefeld University Press. <https://doi.org/10.1515/9783839454190-002>.
- Schonlau, Matthias. 2022. "Patient Joe (Open-ended Question)." GESIS, Cologne. Data File Version 1.0.0. <https://doi.org/10.7802/2474>.
- Selivanov, Dmitriy, Manuel Bickel, and Qing Wang. 2020. *text2vec: Modern Text Mining Framework for R*. R package version 0.6. The CRAN Comprehensive R Archive Network. <https://CRAN.R-project.org/package=text2vec>.
- Silge, Julia, and David Robinson. 2016. "tidytext: Text Mining and Analysis Using Tidy Data Principles in R." *JOSS* 1 (3). <https://doi.org/10.21105/joss.00037>.
- Singer, Eleanor, and Mick P. Couper. 2017. "Some Methodological Uses of Responses to Open Questions and Other Verbatim Comments in Quantitative Surveys." *Methods, data, analyses (mda)* 11, no. 2 (July): 115–134. <https://doi.org/10.12758/MDA.2017.01>.
- Turner, Kenneth J., and Paul S. Lambert. 2015. "Workflows for Quantitative Data Analysis in the Social Sciences." *International Journal on Software Tools for Technology Transfer* 17 (3): 321–338. <https://doi.org/10.1007/s10009-014-0315-4>.
- University of Helsinki. 2014. "UD Finnish-FTB: The UD Version of FinnTreeBank 1." Dataset. <http://urn.fi/urn:nbn:fi:lb-2023050801>.
- Virtanen, Antti, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. 2019. "Multilingual Is Not Enough: BERT for Finnish." ArXiv. <https://arxiv.org/abs/1912.07076>.
- Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the Tidyverse." *Journal of Open Source Software* 4 (43): 1686. <https://doi.org/10.21105/joss.01686>.
- Wijffels, Jan. 2020. *textrank: Summarize Text by Ranking Sentences and Finding Keywords*. R package version 0.3.1. The Comprehensive R Archive Network. <https://CRAN.R-project.org/package=textrank>.
- Wijffels, Jan. 2023. *udpipe: Tokenization, Parts of Speech Tagging, Lemmatization and Dependency Parsing with the "UDPipe" NLP Toolkit*. R package version 0.8.11. The Comprehensive R Archive Network. <https://CRAN.R-project.org/package=udpipe>.
- Wilson, Scott. 2022. *DACT Qualitative Analysis Toolkit (qualkit)*. Python package version 0.1.5. The Python Package Index. <https://pypi.org/project/qualkit/>.

APPENDICES

TEXTRANK ALGORITHM DETAILS

The TextRank algorithm is derived from the PageRank algorithm (Brin and Page 1998) used by Google to rank web pages. This algorithm is based on a directed graph for web surfing. Vertices (V) represent pages and the word importance, or score, $S(V_i)$ for vertex i is determined recursively by:

$$S(V_i) = (1 - d) + d \sum_{j \in In(V_i)} \frac{1}{|Out(V_i)|} S(V_j)$$

where: $G = (V, E)$ is a directed graph with edges $E \subset V \times V$, d is a damping factor (commonly 0.85), $In(V_i)$ are predecessor vertices of vertex V_i , and $Out(V_i)$ are the vertex's successors.

The TextRank algorithm extends PageRank to graphs extracted from text to enable unsupervised keyword and sentence extraction. In the TextRank application of PageRank, an undirected graph is considered where $In(V_i) = Out(V_i)$ and weights of edges are introduced to reflect multiple co-occurrences between vertices. Thus, TextRank introduces a new formula for weighted vertex score:

$$WS(V_i) = (1 - d) + d \sum_{j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{ji}} WS(V_j)$$

with $WS(V_i)$ being the weighted score of vertex V_i , and w_{ji} being the weight of the edge between vertices V_i and V_j .

The TextRank algorithm proceeds as follows: First, text is tokenised into individual words and for each token the relevant part-of-speech (POS) tag is added. Tokens are then filtered based on their POS type. Next, an edge is created between two remaining words if these words co-occur within N words in the text, and all vertices (representing individuals words) are assigned an initial score of 1. The ranking algorithm (in this instance PageRank, although the authors highlight that others could be used instead) is then applied until convergence is reached. Finally, once scores have been obtained for all vertices, these are sorted in descending order and the top T words (commonly a 1/3 of all words) are kept.

LIST OF AVAILABLE LANGUAGE MODELS FOR FINNSURVEYTEXT

The full list of 101 language treebanks available for use with `finnsurveytext` can be found by running `fst_print_available_models()`.

```
R> fst_print_available_models()

[1] "afrikaans-afribooms"      "ancient_greek-perseus"
[3] "ancient_greek-proiel"    "arabic-padt"
[5] "armenian-armtdp"         "basque-bdt"
[7] "belarusian-hse"          "bulgarian-btb"
[9] "buryat-bdt"              "catalan-ancora"
[11] "chinese-gsd"              "chinese-gsdsimp"
[13] "classical_chinese-kyoto"  "coptic-scriptorium"
[15] "croatian-set"             "czech-cac"
```

[17]	"czech-cltt"	"czech-fictree"
[19]	"czech-pdt"	"danish-ddt"
[21]	"dutch-alpino"	"dutch-lassysmall"
[23]	"english-ewt"	"english-gum"
[25]	"english-lines"	"english-partut"
[27]	"estonian-edt"	"estonian-ewt"
[29]	"finnish-ftb"	"finnish-tdt"
[31]	"french-gsd"	"french-partut"
[33]	"french-sequoia"	"french-spoken"
[35]	"galician-ctg"	"galician-treegal"
[37]	"german-gsd"	"german-hdt"
[39]	"gothic-proiel"	"greek-gdt"
[41]	"hebrew-htb"	"hindi-hdtb"
[43]	"hungarian-szeged"	"indonesian-gsd"
[45]	"irish-idt"	"italian-isdt"
[47]	"italian-partut"	"italian-postwita"
[49]	"italian-twittiro"	"italian-vit"
[51]	"japanese-gsd"	"kazakh-ktb"
[53]	"korean-gsd"	"korean-kaist"
[55]	"kurmanji-mg"	"latin-ittb"
[57]	"latin-perseus"	"latin-proiel"
[59]	"latvian-lvtb"	"lithuanian-alksnis"
[61]	"lithuanian-hse"	"maltese-mudt"
[63]	"marathi-ufal"	"north_sami-giella"
[65]	"norwegian-bokmaal"	"norwegian-nynorsk"
[67]	"norwegian-nynorsklia"	"old_church_slavonic-proiel"
[69]	"old_french-srcmf"	"old_russian-torot"
[71]	"persian-seraji"	"polish-lfg"
[73]	"polish-pdb"	"polish-sz"
[75]	"portuguese-bosque"	"portuguese-br"
[77]	"portuguese-gsd"	"romanian-nonstandard"
[79]	"romanian-rrt"	"russian-gsd"
[81]	"russian-syntagrus"	"russian-taiga"
[83]	"sanskrit-ufal"	"scottish_gaelic-arcoosg"
[85]	"serbian-set"	"slovak-snk"
[87]	"slovenian-ssj"	"slovenian-sst"
[89]	"spanish-ancora"	"spanish-gsd"
[91]	"swedish-lines"	"swedish-talbanken"
[93]	"tamil-ttb"	"telugu-mtg"
[95]	"turkish-imst"	"ukrainian-iu"
[97]	"upper_sorbian-ufal"	"urdu-udtb"
[99]	"uyghur-udt"	"vietnamese-vtb"
[101]	"wolof-wtb"	

FURTHER OUTPUT FROM THE EXAMPLE WORKFLOW

The `finnsurveytext` package includes sample data of Finnish survey responses called `dev_coop`, which is from the study [FSD2821 Nuorten ajatuksia kehitysyhteistyöstä 2012](#) (Young People's Views on Development Cooperation 2012) (Finnish Children and Youth Foundation [2019](#)). The top four rows of raw data are shown in [table 2](#).

[Table 3](#) shows the top six rows of formatted `dev_coop` data, which is the output of `fst_prepare()`.

An example survey with English text is included in the package as `english_sample_survey`. This data is available with registration from the GESIS Data Archive for the Social Sciences under a CC BY 4.0 licence (Schonlau [2022](#)). The top six rows of raw data are shown in [table 4](#).

The top six rows of the formatted `english_sample_survey` data are shown in [table 5](#).

fsd_id	q11_1	q11_2	q11_3	paino	gender	region	year_of_birth	education_level
1	1	elämisen tarvittavat perusasiat ovat kehittyvät (esim. vesi, talo, ruoka). Ja niitä ei ole riittävästi	varmaan koitetaan kehittää em	saastuminen ja luonnonvarojen li-ikakäyttö, nälänhätä ja ylikansoittuminen	0.54	Female	Etelä-Suomi	1992
2	2	on kurjuutta ja nälänhätää, asiat eivät ole vielä kehittyneet, lapset eivät pääse kouluun, työillä on huonompi asema kuin pojilla.	pyritään auttamaan?	ihmiskauppa, nälänhätä ja sodat/turvatomuus	0.72	Female	Pohjois- ja Itä-Suomi	1994
3	3	jokaisella ei ole turvattua toimeentuloa ja jossa todella huonaa koulutuksen arvon.	autetaan ja näytetään ihmisille parempaan tulevaisuuteen heidän oman työnsä tulok-sena.	kouluttamattomuus, nälkä ja puhtaan veden puute.	0.62	Female	Helsinki-Uusimaa	1994
4	4	kehityksen taso ei ole yhtä korkea kuin kehittyneissä maissa	yleensä haitataan kehitysmaan kehittymistä	öljy, raha, se fakta että ei olla vielä päästy asumaan muualla kuin tällä yhdellä planeetalla	0.34		Länsi-Suomi	

Table 2: First four rows of data in the *dev_coop* sample data.

doc_id	paragraph_id	sentence_id	sentence	token_id	token	lemma	upos	xpos
1	1	1	saastuminen ja luonnonva- rojen liikakäyttö, nälänhätä ja ylikansoittuminen	1	saastuminen	saastuminen	NOUN	N,Sg,Nom
2	1	1	saastuminen ja luonnonva- rojen liikakäyttö, nälänhätä ja ylikansoittuminen	3	luonnonvarojen	luonnonvaro	NOUN	N,Pl,Gen
3	1	1	saastuminen ja luonnonva- rojen liikakäyttö, nälänhätä ja ylikansoittuminen	4	liikakäyttö	liikakäyttö	NOUN	N,Sg,Nom
4	1	1	saastuminen ja luonnonva- rojen liikakäyttö, nälänhätä ja ylikansoittuminen	6	nälänhätä	nälänhätä	NOUN	N,Sg,Nom
5	1	1	saastuminen ja luonnonva- rojen liikakäyttö, nälänhätä ja ylikansoittuminen	8	ylikansoittuminen	ylikansoittuminen	NOUN	N,Sg,Nom
feats								
			head_token_id	dep_rel	deps	misc	weight	gender
1	Case=Nom Number=Sing	0		root			0.54	Female
2	Case=Gen Number=Plur	4		nmod			0.54	Female
3	Case=Nom Number=Sing	1		conj		SpaceAfter=No	0.54	Female
4	Case=Nom Number=Sing	1		conj			0.54	Female
5	Case=Nom Number=Sing	1		conj		SpacesAfter=\n	0.54	Female

Table 3: First six rows of data in the *dev_coop* formatted data.

	id	label	label_coder1	label_coder2	text
1	1	proactive	proactive	proactive	Joe should talk to the doctor or tell the nurse that the doctor said he has to come back in two weeks.
2	2	proactive	proactive	proactive	I think he should have the receptionist talk to the doctor to make sure that he gets in there at the appropriate time; find out if it actually can be two weeks or if two weeks later would be OK.
3	3	proactive	proactive	proactive	Joe should talk to the doctor and make arrangements to come in in two weeks. He was pretty specific about that.
4	4	proactive	proactive	proactive	I think Joe should insist on an appointment in two weeks.
5	5	proactive	proactive	proactive	Joe should discuss this with the receptionist as to what the doctor told him to do. And insist on seeing him at two weeks.
6	6	proactive	proactive	proactive	He should tell the receptionist to speak with the doctor because he was told to come back in two weeks.

Table 4: First six rows of data in the *english_sample_survey* sample data.

doc_id	paragraph_id	sentence_id	sentence	token_id	token	lemma	upos	xpos
1	1	1	Joe should talk to the doctor or tell the nurse that the doctor said he has to come back in two weeks.	1	joe	joe	PROPN	NNP
2	1	1	Joe should talk to the doctor or tell the nurse that the doctor said he has to come back in two weeks.	3	talk	talk	VERB	VB
3	1	1	Joe should talk to the doctor or tell the nurse that the doctor said he has to come back in two weeks.	6	doctor	doctor	NOUN	NN
4	1	1	Joe should talk to the doctor or tell the nurse that the doctor said he has to come back in two weeks.	8	tell	tell	VERB	VB
5	1	1	Joe should talk to the doctor or tell the nurse that the doctor said he has to come back in two weeks.	10	nurse	nurse	NOUN	NN
6	1	1	Joe should talk to the doctor or tell the nurse that the doctor said he has to come back in two weeks.	13	doctor	doctor	NOUN	NN

feats	head_token_id	dep_rel	deps	misc
1 Number=Sing	3	nsubj		
2 VerbForm=Inf	0	root		
3 Number=Sing	3	obl		
4 VerbForm=Inf	3	conj		
5 Number=Sing	8	obj		
6 Number=Sing	14	nsubj		

Table 5: First six rows of data in the *english_sample_survey* formatted data.