Identifying AI talents among LinkedIn members
A machine learning approach

Thomas Roca, PhD
Microsoft and LinkedIn Economic Graph
Thomas.Roca@microsoft.com

June 7, 2019

Abstract

How to identify specific profiles among the hundred of millions LinkedIn's members? *LinkedIn Economic Graph* thrives on skills, around 50 thousand of them are listed by LinkedIn and constitute one of the main signals to identify professions or trends. Artificial Intelligence (AI) skills, for example, can be used to identify the diffusion of AI in industries [16]. But the noise can be loud around skills for which the demand is high. Some users may add "trendy" skills on their profiles without having work experience or training related to them. On the other hand, some people may work in the broad AI ecosystem (e.g. AI recruiters, AI sales representatives, etc.), without being the AI practitioners we are looking for. Searching for keywords in profiles' sections can lead to mis-identification of certain profiles, especially for those related to a field rather than an occupation. This is the case for Artificial Intelligence.
In this paper, we propose a machine learning approach to identify such profiles, and suggest to train a binary text-classifier using job offers posted on the platform rather than actual profiles. We suggest this approach allows to avoid manually labeling the training dataset, granted the assumption that job profiles posted by recruiters are more "*ideal-typical*" or simply provide a more consistent triptych "job title, job description, associated skills" than the ones that can be found among member's profiles.

1 Introduction

The debate around Artificial Intelligence (AI) induced automation and the labor force is extremely vivid, but rarely based on strong empirical evidence. How fast is AI diffusing in industry and services? Is AI already impacting the job market? A few academic studies propose frameworks to estimate the impact of the fourth industrial revolution on the labor market (Frey and Osborn[7], OECD[3], etc.), but they reach no consensus, providing a wide range of outcomes. The public debate is fueled with guesstimates filling the vacuum left by academia. Indeed, data is scarce when it comes to Artificial Intelligence workers and actual implementation of AI systems within companies. Tech enthusiasts argue that AI will empower workers, complementing their skills and allow people focusing on analysis, critical thinking, ingenuity or human to human interactions. The truth is we have no comprehensive statistics that describe AI diffusion within companies and societies and current estimates are based on qualitative data and perception surveys from astonishingly small samples.
What's more, the European Union and governments across Europe are currently devising their own AI strategy, craving for more objective data and insights about AI skills needed, AI diffusion progress and hindrance.
We suggest LinkedIn insights can provide answers to some ofthese questions. By scrutinizing AI talents, their skills, the industry they work for, but also the vacancies advertised on

our platform we can provide the first building block of a more comprehensive piece of research to better inform the debate about the impact of AI on the Labor market.
Identifying AI specialists among the hundreds of petabytes of LinkedIn database reveals a "needle in a haystack" type of problem. To distinguish AI specialists who actually implement AI, from other members with a mention of AI in their profile, I built a text classifier using Scikit-learn and NLTK. The model was trained using AI-related job offers data from different countries. This paper describes the methodology I developed and the challenges I faced to transform LinkedIn big data into actionable statistics for policy makers. The study covers 44 countries[1] and a data-visualization portal allows making the best out of these results.

1.1 Related work

Estimates of the AI worker population, can be found on the web, but the methodology used to construct those numbers are rarely detailed or when they are, they duly acknowledge they are working assumptions, based on existing biased data sources.Chinese tech giant Tencent, published the *2017 Global AI Talent White Papers* [19] estimating the AI practitioners' population to approximately 300.000 individuals[21]. In 2017, *The New York Times*, citing Jean-François Gagné, from *Element AI*, used the estimates of approximately 10.000 AI experts worldwide[14]. This estimates is the one taken by the European Commission's *European Political Strategy Center* in its March 2018 Strategic Notes *The Age of Artificial Intelligence, Towards a European Strategy for Human-Centric Machines*[5].

Since then, Element AI has published the *Global AI Talent Report 2018*[8] in which the AI talents population estimates were updated to 30.000 AI experts, based not only on academic publication and conferences but also based on crawling LinkedIn's website. Element AI, explains clearly the limitation of their estimates and the biases of the data sources, for instance, the over-representation of the English speaking and western-world both in LinkedIn population and academic conferences.

Aware of those limitations, our research does not seek to provide an estimate of AI talents worldwide. Although our machine learning model does come with an estimate for this population - and score associated with this prediction. Our objective is simply to identify the largest possible sample of AI talents.

On the demand side, literature exists analyzing the emergence of data-science and Artificial Intelligence in the software and IT industry. Qualitative research such as the one done by Kim, M. et al. in 2016 [10] help understanding the demand for those relatively new roles and the skills and background needed for such positions, but also the motivation of data-scientists. Estimates of AI related job offers are easier to find as companies have been specializing in scrapping the web to identify *Data Science and Analytics* job offers. For example, *Bruning Glass, IBM and BHEF* published in 2017 a report[13] analyzing such job offers from various sources and provide comprehensive insights about the magnitude of the demand in the United States, the type of roles and their requirements and salaries for those positions. Nevertheless, this report has a broader scope than our analysis of the demand side as it includes data analytics and engineering. It has also a more limited breadth as our study covers not only the demand but also the supply side and goes beyond the United States.

Drawing upon research on AI skills done by LinkedIn Economic Graph[2] we propose a new framework to provide a sound and comprehensive description of the AI labor force: the skills AI talents own, the industry they work in, their education, etc. By

[1] Argentina; Australia; Austria; Belgium; Brazil; Bulgaria; Canada; Chile; Colombia; Croatia; Cyprus; Czech Republic; Denmark; Estonia; Finland; France; Germany; Greece; Hong Kong SAR China; Hungary; Ireland; Italy; Latvia; Lithuania; Luxembourg; Malta; Mexico; Netherlands; New Zealand; Norway; Philippines; Poland; Portugal; Romania; Singapore; Slovak Republic; Slovenia; South Africa; Spain; Sweden; Switzerland; Thailand; United Kingdom; United States.

[2] see How artificial intelligence is already impacting today's jobs

scrutinizing AI related job offers we were also able to identify skills gap between demand and supply to provide a complete picture of the AI labor force.

To our knowledge, there is currently no academic literature covering this specific topic. However, the methodology we implemented in the field of *Natural Language processing* is well covered by computer science literature. For surveys on Natural Language Processing and machine learning models for binary text classification see [11], [9], [20], [12] and [2].

1.2 Identifying AI talents in LinkedIn database: a machine learning approach

Artificial Intelligence is not a job title, it is a field. There is no such standardized category within LinkedIn database. Thus, to identify AI practitioners among all our members we need to scrutinize their skills and employment. Querying LinkedIn database with keywords such as "Artificial Intelligence", "Machine learning", "Deep learning", we would capture not only AI talents but also, sales people specialized in selling AI technology, recruiters, public speakers and advocates, and many variations of those profile we simply cannot foresee. Indeed, it is difficult to come up with a clear set of rules to properly query the database to identify AI talents, the ones who actually implement AI.

Machine learning is usually a good approach when confronted to a "Needle in a haystack" type of problem. My assumption is that exposing a text classifier to enough AI talent profiles, I will be able to *filter and refine the results of an initial and more generic query for AI talents on LinkedIn database*. For this study I used python 3.5, Scikit-Learn and the NLTK library. The quality of predictions of machine learning models primarily lies in the quality of the data used to train them. The next section will describe the approach I used for building a proper training dataset.

2 Building the training dataset

Manually labeling thousands of profiles proved extremely time consuming, and potentially leads to encoding our own biases in the model. After having started with this approach, and scrutinized the skills, job title and description of the most prominent AI workers, I decided to resort to AI vacancies, advertised on the platform. The job database contains job titles, position description and the skills associated with it. Our assumption is that the jobs (title, description, skills) containing AI-related keywords will likely provide an accurate description AI talents' profiles while containing less noise than LinkedIn members' profiles. I assume that the very nature of job advertising data allows to use string search for building a relatively pure training dataset and that by harvesting enough of those, my training dataset will contain enough variation to be able to capture most of AI talents in their respective area of expertise (computer vision, Natural language processing, robotics, knowledge reasoning etc.). I thus built "synthetic profiles" gathering job titles, position descriptions and skills extracted from the job table[3] using a string search for: ["Artificial Intelligence", "AI", "Computational Intelligence", "Reinforcement Learning" , "Machine Learning", "Deep learning", "neural networks"] - and their translation in local languages when needed.

However, to be able to detect non-AI practitioners in the member database, I need to have similar categories in my training dataset. Therefore, I also need to find jobs advertising for non-AI practitioners which, nevertheless, contain certain AI-related keywords. To identify those "false positive", I looked[4] into jobs' title and checked if, besides AI related keywords, they also contained

[3] For USA, Great Britain, Australia since January 2017 for the "AI class".

[4] For USA, France, Great Britain, Germany, Australia since June 2016 for the "Non-AI class". It is more challenging to find non-AI practitioners in the Job database when querying for AI keywords, for this reason, I expanded both time and space coverage for this class.

one of those: ["account manager", "practice leader", "Account Manager", "Sales", "Practice Leader", "Product Manager", "Recruiter", "Business Analyst", "Sales Engineer", "Evangelist", "Representative", "Digital Analyst", "human resource", "customer success", "speaker"]. These keywords are the ones I identified as the most prominent in non-hard-skilled AI related profiles. I assumed that overall both categories, "AI" and "non-AI", job offers will provide enough variation and help building a sound binary classifier.

2.1 Standardization issues and users' supplied information

Standardization in a free text environment is a huge challenge. Standardization coverage varies by language – standardization is strongest for English language profiles. To make sure to grasp as much information as possible, I decided to use both standardized and user supplied information. I resorted to *Microsoft Bing translation API* to automate translation of resulting synthetic profiles. Using both standardized and user supplied information I must make sure no duplicated information remains in the final profiles.

2.2 Data preparation and ingestion

Models do not understand words, we need to transform text into numbers, extract features that characterize text input (member's information). Several technics exist to extract features from text. Common features extraction methods consist in counting (Count Vectorizer aka bag of words) unique words or counting words next to each other (N-gram). It is also possible to count the frequency of words (Terms Frequency Inverse Document Frequency - TFIDF), the amount of punctuation, capitalized letters, length of the text (use case e.g.: spam detection). When features are extracted from raw text, distinctions are made between each and every character: space, coma, uppercase, punctuation, can thus end up influencing the outcome, although in our case, those do not carry meaningful information for distinguishing AI from non-AI practitioners. To reduce the noise in our texts and prevent increasing the number of features, I followed those steps:

- Text passed to lower case;
- Stop words were removed (see NLTK stop words);
- Punctuation and special characters were removed;
- Job description turned into keywords (using *Rake library)*.

With 30% of "Not AI" profiles in the training dataset, our training set is a bit unbalanced, extra-care will be needed while evaluating the models. Confusion matrices are a good way to evaluate unbalanced misclassification. See figure 3 on page 11.

3 Binary text classification with Scikit-Learn

3.1 Vectorizers and models benchmarking

Deep neural networks are gaining momentum in text classification. TensorFlow proposes a text classifier using deep neural networks but I ultimately decided not to resort to deep learning to allow better explainability of our results (differences in performances were small enough to feel good with this trade-off). One of the main differences between machine learning and deep learning, is the control over feature extraction. With deep learning, the model also chose the best way to extract features while in traditional machine learning, the developer has control over it. This has important consequence on model interpretability and transparency.
Support Vector Machine, together with *Logistic Regression*, are the go-to models for binary classification in supervised machine learning settings. SVM draws its name from the data points (the support vectors) which support the margins between which is drawn the hyperplane (i.e. they are the data points the closest to the other class, the hardest to classify). Logistic Regression operates to maximize the

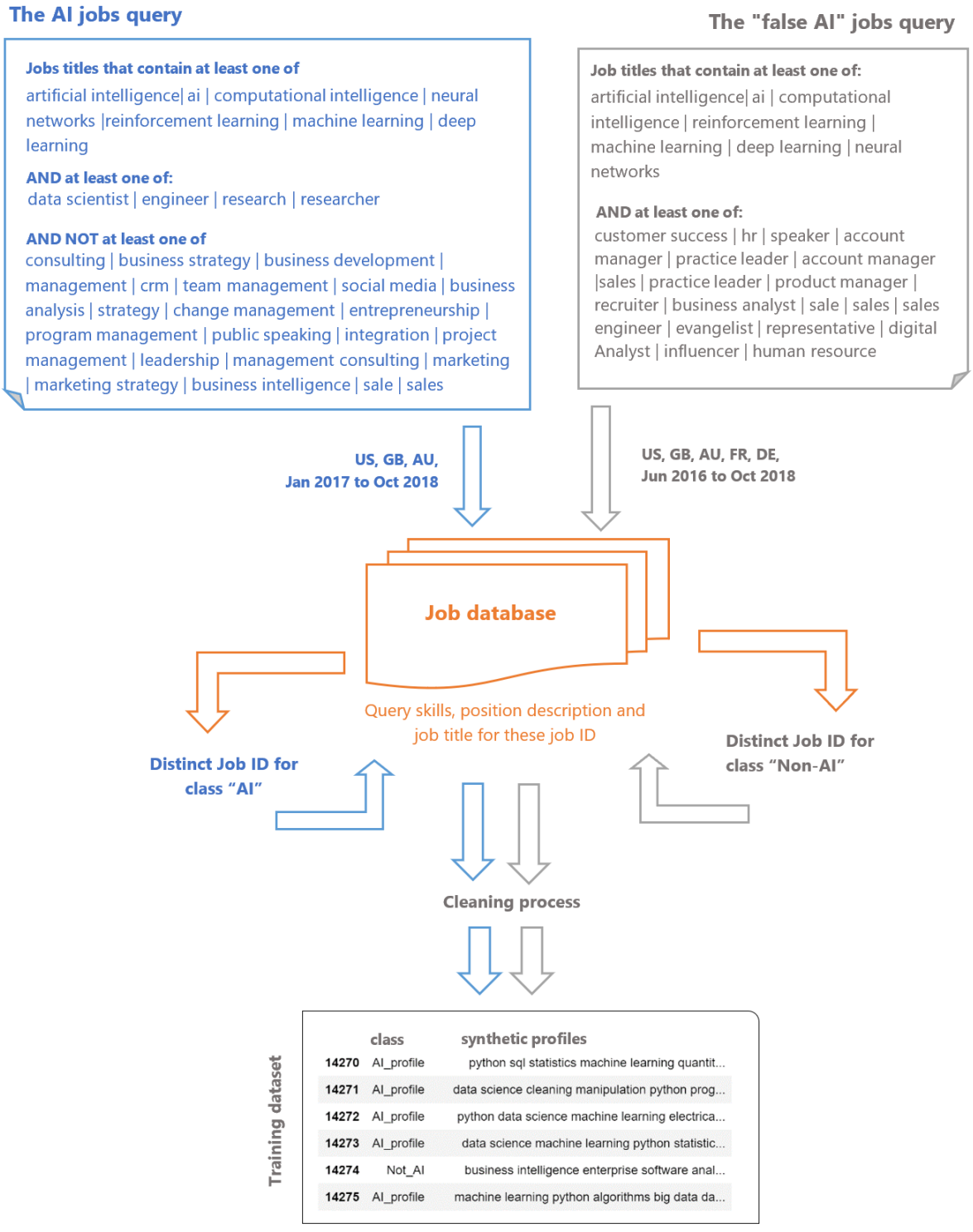

Figure 1: Flow chart: building the training dataset

likelihood that a given data point is well classified (i.e. Maximum Likelihood Estimation). SVM properties are considered better suited for generalization and scalability [18], while LR provides "calibrated probabilities that can be interpreted as confidence in a decision"[18]. *NB. We will use this feature of LR in Platt calibration, see sub-section 4.3.* We used a linear Support Vector Classifier which is a generalization of SVM.

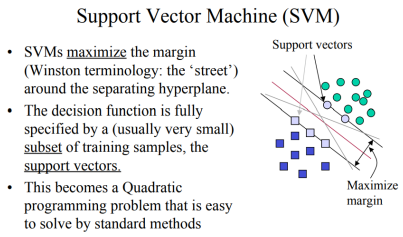

Figure 2: SVM explained [4]

For a discussion on Support Vector Machine vs. Logistic Regression, refer to [18], [6], [15] and [4]

3.2 Testing Vectorizers

I tested both TFIDF and Count Vectorizers as features extractors and different models (Logistic Regression, Random forest, Naïve Bayes, Perceptron, Support Vector Machine, etc.). Our intuition that bags of word would provide an efficient vectorizer was confirmed (tradeoff execution time / performance). Indeed, the nature of the data, mostly list of words (skills, job title and Key words extraction for job description) renders less necessary the weighting procedure used in TFIDF and will save us a fair amount of computing time. Performance wise, the bag of words approach even provides a slightly better accuracy for selected models. A Support Vector Machine Model (Support Vector Classifier with linear kernel) with bags of words vectorizer provided the best results with 99.7% accuracy on our testing set- figure 5, page 12 - (8 synthetic profiles misclassified), slightly outperforming Logistic Regression - see figure 6, page 12

3.3 Cross Validation

For training and testing purposes the training dataset is split into two subsets. I use a 20/80 ratio (i.e. 20% of the training set is kept for testing). To benchmark the performance of alternative models and make sure the results are consistent across all the dataset, I performed a cross-validation analysis using 5 iterations. This procedure compares the performances between the different models using new randomly generated testing/training set for each iteration - see figure 9 on page 14. Note that the testing set contains 20% of the training dataset (i.e. 2858 observations).

For me information on SVM and implementation with Scikit-learn see [1]

3.4 Tuning Hyper parameters

When fitting a Support Vector Machine classifier, two hyper-parameters are considered important: the *Kernel trick* and the *Penalty parameter C of the error term.* The first allows to specify whether data are linearly separable and the second allows fine-tuning how closely the classifier fits to the training data. To fine-tunes those two parameters I used *GridSearchCV*, the outcome shows that the C parameters had little influence on the results and that the data are linearly separable - see figure 7 on page 13. As a result, I kept the default value for the C parameters and used the linear kernel.

3.5 Most informative features

Inspecting the most informative features driving the classification, we can observe how the model behaves and correct for features we do not want - for example company names. See figure 8 on page 13.

4 Implementation

Once the model has been trained, I can use it to classify actual profiles (from *pseudonymised* and concatenated profiles sections: title, current or last position description, skill set). For the AI in the Labour Market series, I attempt to build AI talents pool for 44 countries.

4.1 Data ingestion

The classifier described in this paper is designed to distinguish AI talents from a subset of members who possess those "AI keywords" we mentioned before. I started with a broad keywords search on the database for the given countries. This query should be broad enough, both specific and generic. To build this query I looked at specific AI libraries: keras, pytorch, scikit-learn, tensorflow, theano, CNTK, caffe (taking case and spelling variations into account). I also used generic terms referring to AI: "artificial intelligence, computational intelligence, reinforcement learning, machine learning, deep learning, neural networks" and their translation in local languages. NB. These keywords are the results of an AI taxonomy effort led by the Economic Graph team. A member ID will end up in my bucket if at least one of those keywords appears in one of those profile sections: position summary, user supplied title, std skill name, user supplied skills. This bucket will contain, non-AI practitioners, sales people, HR, AI advocates, etc. that's where the text classifier comes in, to distinguish AI from not-AI practitioners. NB. For jobs position of our members, I looked at "final active" or "current position", not their complete work history.

4.2 Prepare the data for prediction

Once collected the skills, position description and title of potential AI talents, this data must be cleaned/prepared to feed the classifier. I translated user-supplied information into English using *Bing translator API*. Duplicates between standardized and user supplied skills were removed, so as stop words and non-alphanumerical characters. For position summary, I also used keywords extraction (Rake library) and translated this into English. At the end of this process, I end up with synthetic profiles: concatenation of title, last/current position description (as keywords), skill set and corresponding member IDs.

4.3 Prediction

Binary classification sorts the population into two groups. Support Vector Machine, in particular, classifies observations into two groups taking the value +1 or -1. SVM does not come with a probability or certainty degree metric. Nevertheless, we can resort to *Platt calibration* to compute the probability distribution associated with each classes. Platt calibration was initially designed by John C. Platt for SVM. It uses a logistic regression model to "map the SVM outputs into probabilities" [17] the resulting probabilities can be used as confidence levels. In Scikit-learn you can pass this option in the parameter of your SVC: *SVC(probability=True, kernel='linear')*. More information on model calibration are available in Scikit-learn documentation. NB. Scikit-learn relies on liblinear libray for linear SVC which uses a *Coordinate descent* as optimization program.

Once this score computed, I decided to use a threshold of 0.95, excluding from the AI talents pool all observations for which the prediction score is below 0.95. This is arbitrary, and if I were to provide raw count, I would provide it as a range with associated scores. However, this is not my attempt. For this study, the threshold discussion is not so crucial. The AI talents series will only provide, aggregates and rankings that should be consistent for a wide range of thresholds. Furthermore, the score distribution is clearly left-skewed. See figure 12 on page 15. To put it differently, in this classification problem we are more interested in precision than recall, we aim at building a sample of population as representative of AI talents as possible, even if it implies that *true-positives* close to the threshold are excluded.

5 Discussion

Our objective was to *go beyond keywords* to refine a pre-selection of potential AI talents extracted from our database using generic AI-related keywords. In this subset we identified that many of resulting potential AI talents were actually not AI practitioners. Our intu-

ition was that a machine learning model can identify better the features able to distinguish between "AI practitioners" and "Not AI practitioners". However, I resort to keywords to build my training set and the method I propose only has a value added *if the features generated by the classifier capture a different subset of members than a simple query on the database searching for the keywords that help building the training set.* First distinction to make here, is that the training set was build not using members' profiles but job offers advertized on the platform. Although vacancies contain similar sections to members profiles (skills, job title, job description), the way they are formulated are different, job offers tend to provide a more "consistent" triptych: job title, job description and associated skills.

Would the keywords used to build the training set be good a predictor of the subset identified by the classifier on the top of the broader initial query? Probably as overlaps exist, some of the *most informative features* used by the model naturally reflect those keywords. But the features built by the model and influencing the classification go way beyond the few keywords used to select the job offers that feed the training set. Furthermore, the weights of those features, their interplay, pushing in both directions to make a final prediction would be very difficult to explicit and encode for example in a rule based algorithm. Figure 11, page 15 shows how different are the resulting talent pools when identified via the keywords used to build the training set or via our SVM classifier.

Our intuition was also that a machine learning model would be easier to implement, less arbitrary and more agile as the features identified by the model also reflect the labor market and will evolve with it when reproducing the study. Indeed, the way to characterize AI practitioners evolves overtime, as the industry evolves. More agile also because using a predictive model, we can easily play with the threshold (the confidence interval) to loosen the criterion and allow the "net" to be wider and capture a broader population. This is particularly interesting for us at LinkedIn, as we would like to identify less specialized members who could be trained and up-skilled to meet to the demand for AI talents.

Ultimately, the metric we lack to properly evaluate the robustness of the method is the actual accuracy of the model when classifying members' profiles, but for this, we would need enough of labeled profiles - and if we had those we would simply train the model this way.

6 Descriptive statistics and visualization

Once identified AI talents, we can get more information about them: skills, industries, inferred seniority, gender, location, company, education, diplomas, etc. As mentioned previously, LinkedIn Economic Graph policy is to not publish raw counts. Considering our methodology, raw counts would only be a prediction range with associated confidence score. For this research, we simply want the best sample possible from which to construct descriptive statistics. I computed those and built charts and maps (using leaflet and Highcharts), such as AI talents' repartition by administrative level[5], by Industries[6], seniority groups, gender, skills, universities, companies, etc. Averages for the European Union and worldwide average were computed using weighted averages (using the share of AI talents by countries). Those insights will be gathered on a web portal hosted on *Azure* to help policy makers make the best of them.

6.1 Data Limitations and things to keep in mind when analyzing those results

- LinkedIn's market penetration varies from one country to another. Both in terms of membership and jobs advertizing. In some countries the number of AI related job offers for the considered period can be small (this should be taken into account when comparing countries).

- Gender balance: not all countries have the same gender representation among

[5] See figure 13 on page 16
[6] See figure 14 on page 16

LinkedIn members. In some countries women can outnumber men on LinkedIn, and vice versa.

- Geographic information: not all countries have the same geographic information associated with members

The AI talents in the Labor Market series only assess Artificial Intelligence prevalence within LinkedIn's population. Given LinkedIn's strong presence in the IT and computer science community, the information we share as rankings, distributions and aggregates are likely reliable representations of the AI labour force for the given countries. Nevertheless, we do not claim statistical representativeness of our sample. These results shall be understood as estimates derived from LinkedIn population.

7 Conclusion

Using Artificial Intelligence vacancies published on LinkedIn, we were able to build *synthetic profiles* describing AI talents and train a binary text classifier to distinguish AI practitioners from a broader population of AI related profiles extracted from LinkedIn database. Doing so, we were able to save a considerable amount of time not having to manually label profiles to build our training dataset. This approach could be used to identify less specialized members who could be trained and up-skilled to meet to the demand for AI talents, or even scale-up and be used to eventually match different kinds of job offers and talent pools on LinkedIn. For the later, the remaining challenge would be to properly label the "not what we are looking for class". A track to explore could consist in building a user interface to propose recruiters to identify a few types of profiles they would like to avoid and within a few iterations constructing this second class.

References

[1] 1.4. Support Vector Machines — scikit-learn 0.20.3 documentation. 6

[2] Charu C. Aggarwal and Chengxiang Zhai. *A Survey of Text Classification Algorithms*. 3

[3] T. Gregory Arntz, M. and U. Zierahn. The risk of automation for jobs in oecd countries: A comparative analysis, 2016. 1

[4] R Berwick. An Idiot's guide to Support vector machines (SVMs). page 28. 6

[5] European Political Strategy Center. The age of artificial intelligence, towards a european strategy for human-centric machines, 2018. 2

[6] Tristan Fletcher. Support Vector Machines Explained. page 19, 2008. 6

[7] Carl Benedict Frey and Micheal A. Osborne. The future of employment: how susceptible are jobs to computerization?, 2013. 1

[8] J.F. Gagné. Global ai talent report 2018, 2018. 2

[9] Rajni Jindal, Ruchika Malhotra, and Abha Jain. Techniques for text classification: Literature review and current trends. page 28. 3

[10] Miryung Kim, Thomas Zimmermann, Robert DeLine, and Andrew Begel. The emerging role of data scientists on software development teams. In *Proceedings of the 38th International Conference on Software Engineering - ICSE '16*, pages 96--107, Austin, Texas, 2016. ACM Press. 2

[11] Roshan Kumari and Saurabh Kr. Machine Learning: A Review on Binary Classification. *International Journal of Computer Applications*, 160(7):11--15, Feb. 2017. 3

[12] Dapeng Liu, Virginia Commonwealth, and Yan Li. A Roadmap for Natural Language Processing Research in Information Systems. page 10. 3

[13] Braganza S. Markow, W. and B. Taska. The quant crunch. how the demand for data science skills is disrupting the job market, 2017. 2

[14] C. Metz. Tech giants are paying huge salaries for scarce a.i. talent, 2017. 2

[15] Andrew Ng. Cs229 lecture notes: Support vector machines. 6

[16] Igor Perisic. How artificial intelligence is already impacting today's jobs, 2018. https://economicgraph.linkedin.com/blog/how-artificial-intelligence-is-already-impacting-todays-jobs. 1

[17] John C. Platt. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In *Advances in Large Margin Classifiers*, pages 61--74. MIT Press, 1999. 7

[18] Kevin Swersky. Support Vector Machines vs Logistic Regression. *Logistic regression*, page 23. 6

[19] Tencent. 2017 global ai talent white papers, 2017. 2

[20] M. Thangaraj and M. Sivakami. Text Classification Techniques: A Literature Review. *Interdisciplinary Journal of Information, Knowledge, and Management*, 13:117--135, 2018. 3

[21] The Verge Vincent, J. Tencent says there are only 300,000 ai engineers worldwide, but millions are needed, 2017. 2

8 Appendix

```
Class: ['AI_profile', 'Not_AI']
Frequency: [69.90481522956327, 30.095184770436727]
Actuall Count: [9988, 4300] | Total: 14288
```

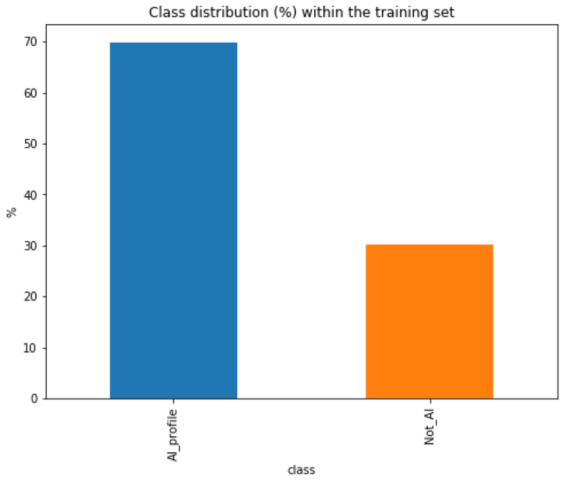

Figure 3: Class distribution in training dataset

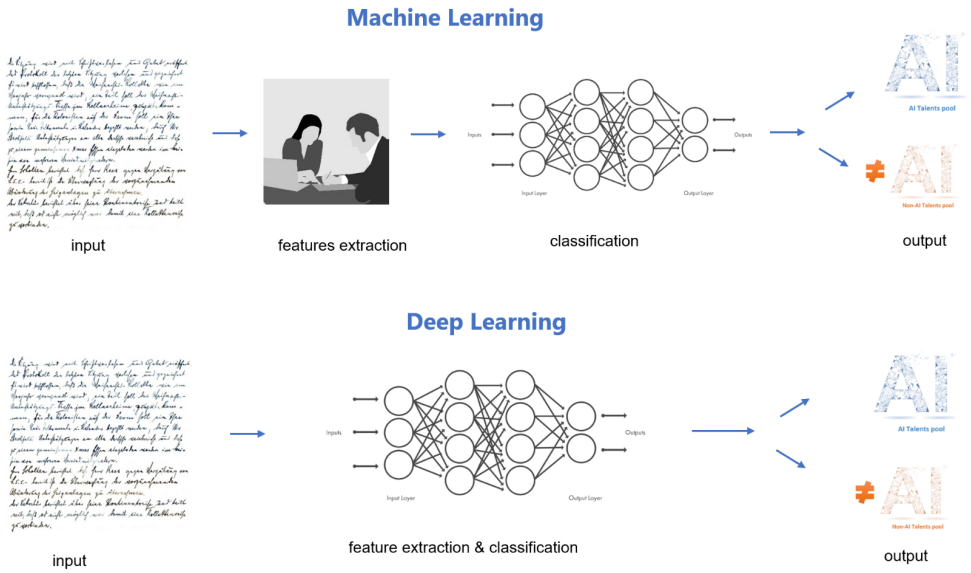

Figure 4: Machine learning versus deep learning processes

```
accuracy:   0.997
Confusion matrix, without normalization
```

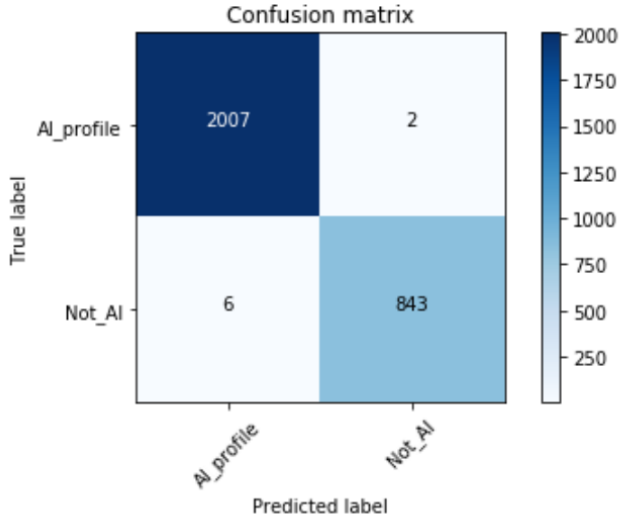

Figure 5: Confusion matrix: Linear Support Vector Machine with Bags of Words vectorizer

```
Logistic Regression
accuracy:  0.993
Confusion matrix, without normalization
```

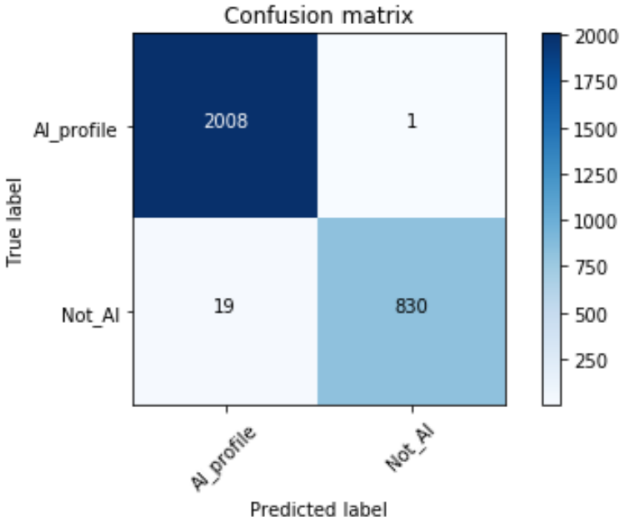

Figure 6: Confusion matrix: Logistic Regression with Bags of Words vectorizer

```
Best Params:{'C': 0.1, 'kernel': 'linear'}

0.996 (+/-0.0) for {'C': 0.1, 'kernel': 'linear'}
0.918 (+/-0.017) for {'C': 0.1, 'kernel': 'rbf'}
0.996 (+/-0.0) for {'C': 1, 'kernel': 'linear'}
0.979 (+/-0.008) for {'C': 1, 'kernel': 'rbf'}
0.996 (+/-0.0) for {'C': 10, 'kernel': 'linear'}
0.991 (+/-0.004) for {'C': 10, 'kernel': 'rbf'}

Best estimator:
SVC(C=0.1, cache_size=200, class_weight=None, coef0=0.0,
  decision_function_shape='ovr', degree=3, gamma='auto', kernel='linear',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False)
```

Figure 7: Fine Tuning hyper parameters

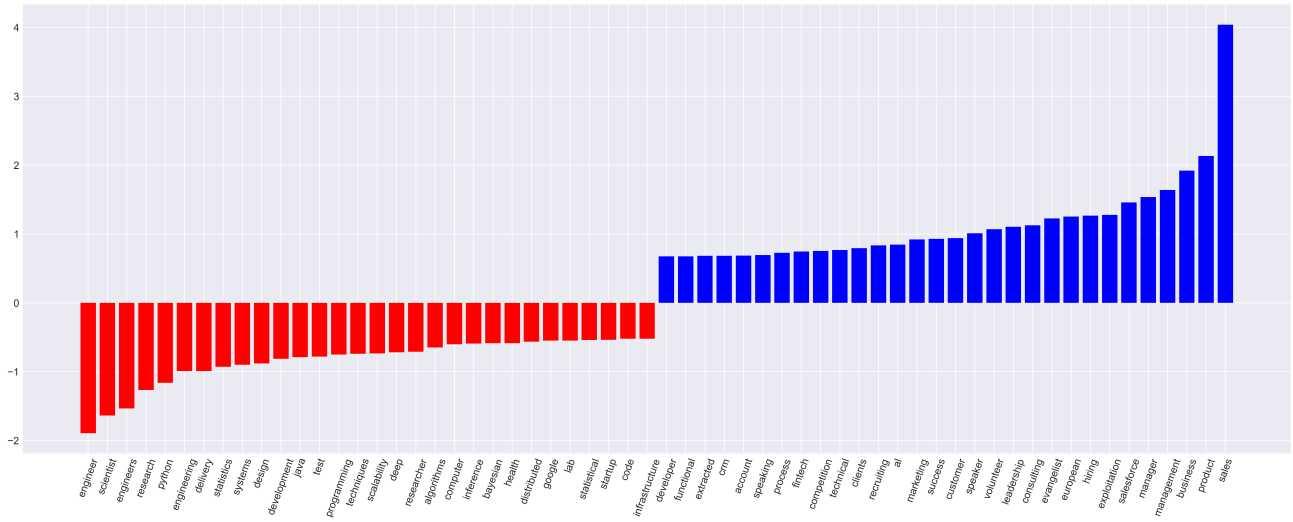

Figure 8: Most informative features

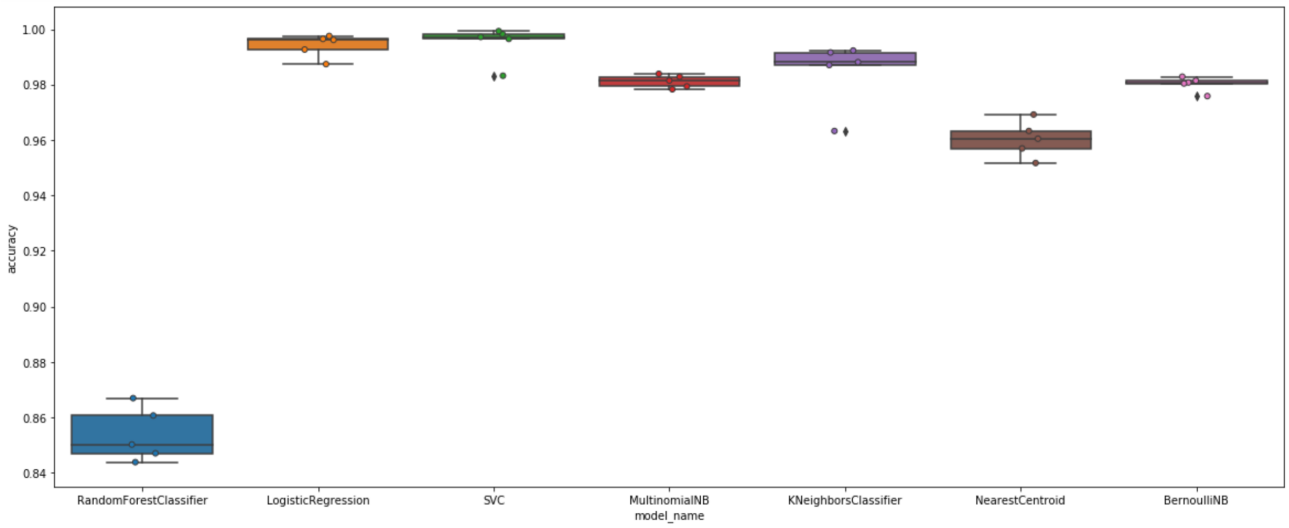

Figure 9: Benchmarking and Cross validation

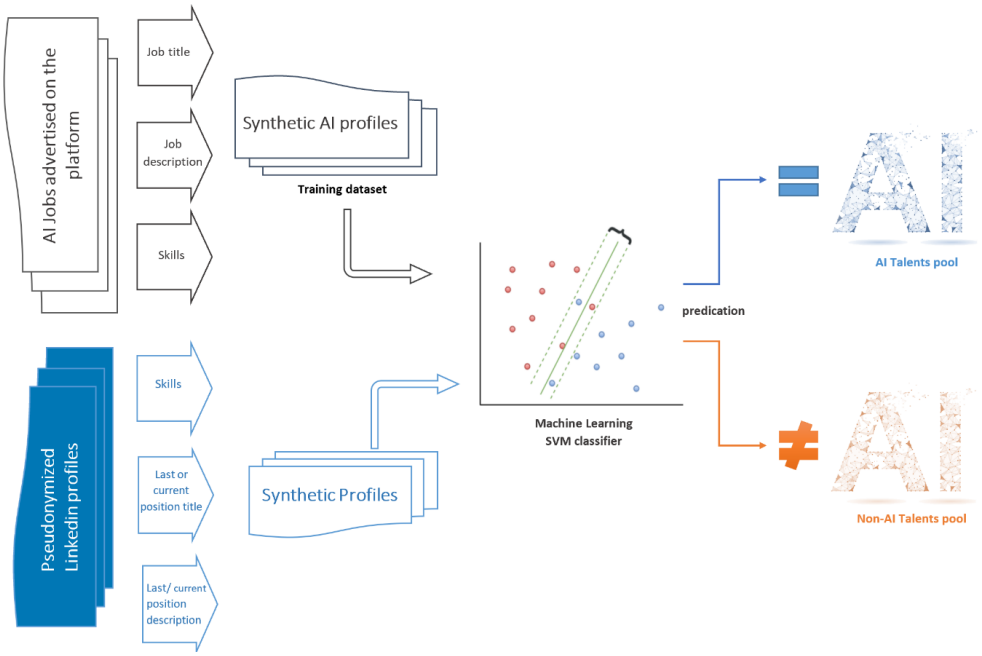

Figure 10: Flow chart training the classifier

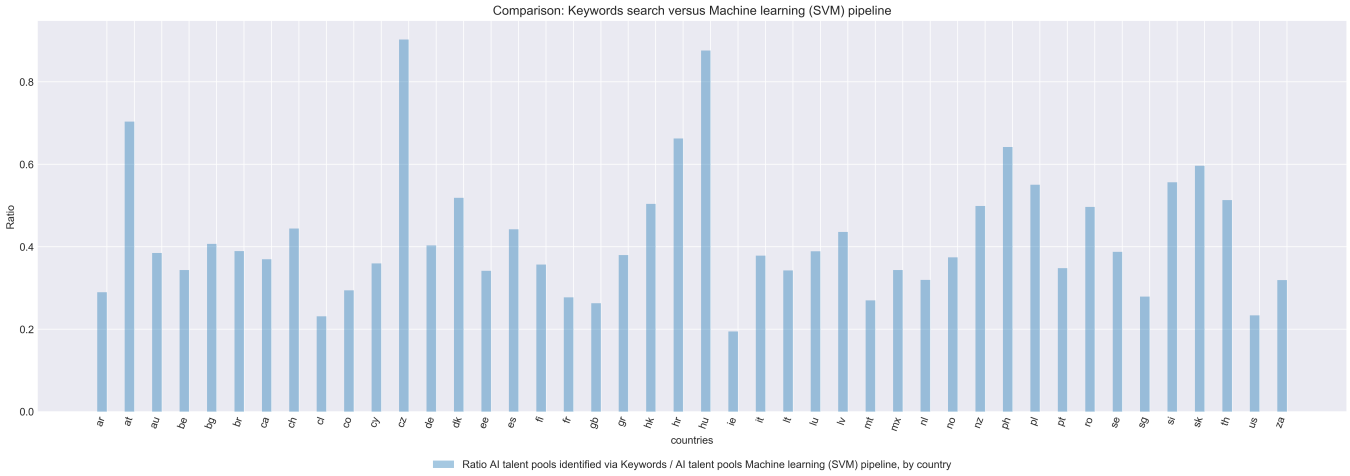

Figure 11: Comparing the size of AI talents pools identified with alternative methods: ratio Keywords search vs. Machine learning (SVM)

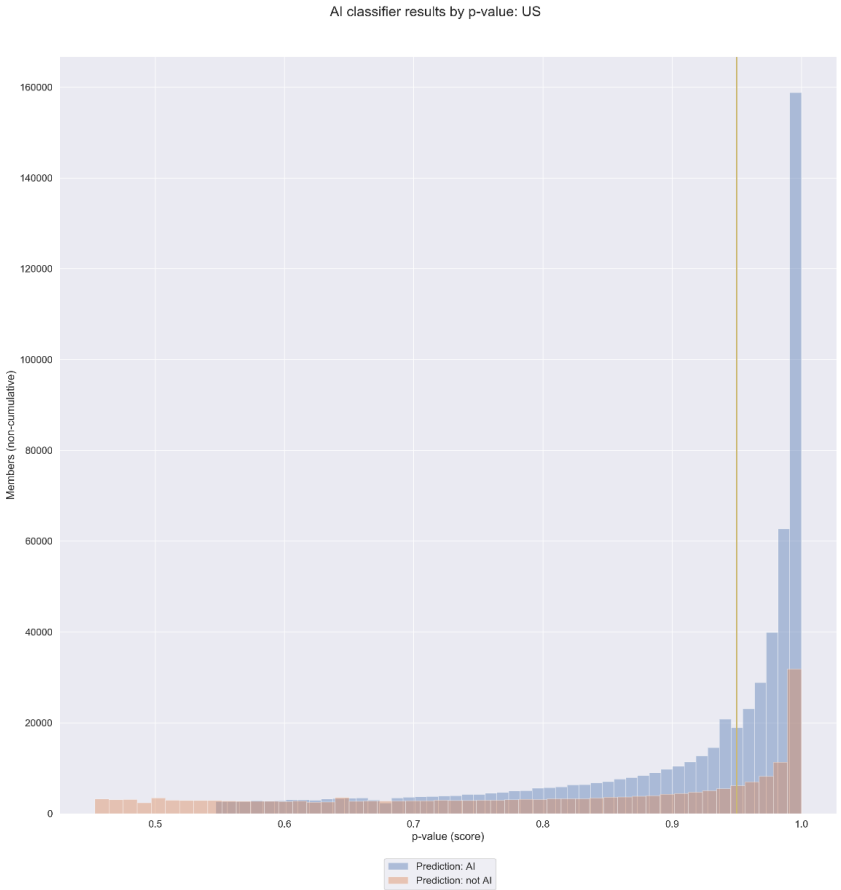

Figure 12: Score distribution for Prediction of AI talents in the US

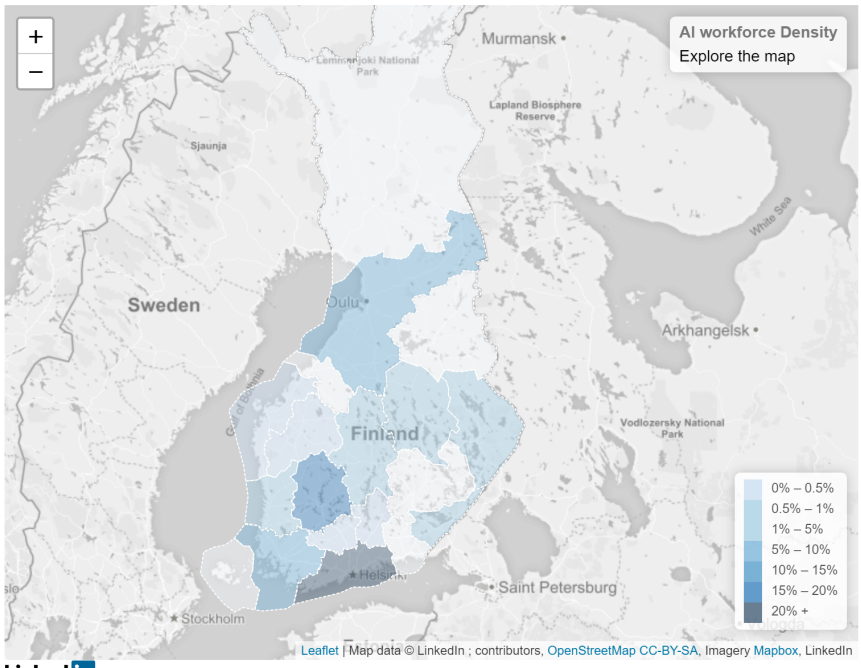

Figure 13: AI talents distribution by administrative levels in Finland

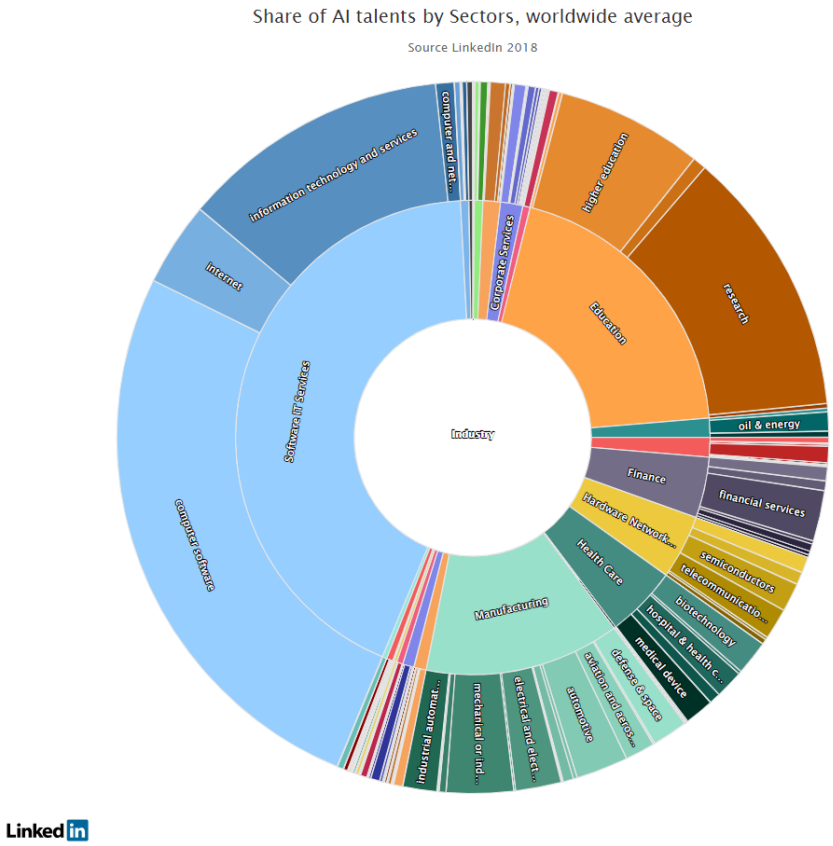

Figure 14: AI talents distribution by industries