Detecting Trends in Job Advertisements

Khalil Mrini[1], *Kshitij Sharma*[1 2], *Pierre Dillenbourg*[1]

[1]Ecole Polytechnique Fédérale de Lausanne, Switzerland
[2] Faculty of Business and Economics, University of Lausanne, Switzerland
{khalil.mrini, kshitij.sharma, pierre.dillenbourg}@epfl.ch

Abstract

We present an automatic method for trend detection in job ads. From a job-posting website, we collect job ads from 16 countries and in 8 languages and 6 job domains. We pre-process them by removing stop words, lemmatising and performing cross-domain filtering. Then, we improve the vocabulary by forming n-grams and restrict it by filtering based on named-entity and part-of-speech tags. We split the job ads to compare two time periods: the first halves of 2016 and 2017. A trending word is defined as a word with a higher TF-IDF weight in 2017 than in 2016. The results obtained show a close correlation between the position of a word in its text and its trendiness regardless of country, language or job domain.

Index Terms: Trend Detection, Keyword Extraction, Cross-Domain Filtering, Training Needs, Text Mining, Natural Language Processing

1. Introduction

In the business of affecting human performance, the careful examination of a problem or system in order before committing training resources is called Training Needs Assessment [1]. It has been described as "*fundamental to the success of a training program*" [2]. This concept applies as well in education, where universities attempt to adapt their curricula to the trends of the job market demand.

Detecting trends has become a recurrent problem with the rise of social media. Cataldi et al. [3] propose a trend detection algorithm on Twitter based on user authority and content aging. Kämpf et al. [4] propose a detection of emerging trends using Wikipedia traffic data and the links between Wikipedia articles. Zhang et al. [5] automatically extract keywords from documents using conditional random fields, but their method needs labelled data.

In this paper, we propose an automatic method for trend detection in job advertisements. First, we collect job ads from a job-posting website, in different countries, languages and job domains, and pre-process the job ad descriptions. Then, we improve and filter the vocabulary, before applying our trend-detecting method on unlabelled data. Finally, we illustrate our results by the trending words obtained in the UK's Information Technology job ads.

2. Data Collection

In this section, we describe the dataset that was collected for the task and the first pre-processing steps that were performed on it.

2.1. Dataset

The job ads used in this paper were collected in mid-June 2017 on Adzuna[1], an online search engine for job ads, using their official API. They offer job ads in 16 countries and in 8 languages:

- Dutch (the Netherlands)
- English (Australia, Canada, India, New Zealand, Singapore, South Africa, the United Kingdom, the United States)
- French (France)
- German (Austria, Germany)
- Italian (Italy)
- Polish (Poland)
- Portuguese (Brazil)
- Russian (Russia)

The job ads are classified in many categories, from which 6 were selected:

- Accounting and Finance
- Consultancy
- Engineering
- Information Technology
- Public Relations, Advertising and Marketing
- Science and Quality Assessment

The ads that were collected were the ones that remained active at the time of collection. Each ad contains a title, a date of posting, a truncated description containing the first 500 characters of the original one, and the URL leading to the source website. If the URL was accessible and the beginning of the truncated description was found in the text it contained, the description used will be the one from the source website.

The trend-detecting pipeline will be applied to each job domain in each country separately.

Job ad descriptions are noisy and keywords that designate skills or other concepts of interest are hard to find. They contain sometimes sentences not related to the job but presenting the company, or sentences to encourage applications.

2.2. Pre-processing

Three steps were used for pre-processing: stop words removal, lemmatisation and cross-domain filtering.

[1]Adzuna is accessible on: https://www.adzuna.com/

2.2.1. Stop Words Removal

The stop word lists were the ones in the corpus [6] provided by the Natural Language Toolkit (NLTK) [7]. They are available in 11 languages, including all of the languages in which the job ads are. We use an automatic language detection algorithm to detect which language's stop word list has the most occurrences with a given collection of job ads.

2.2.2. Lemmatisation

To diminish noise in the job ads, we use lemmatisers to get conjugated verbs, plural nouns and other inflected words to their uninflected form. The WordNet lemmatiser [8] was used for English. Lemmatisers for Dutch, French, German and Italian were extracted from the Pattern package [9]. We also used a Yandex algorithm for Russian lemmatisation [10], a Polish inflectional dictionary [11] and a Brazilian Portuguese lemmatiser[2] based on a Maximum Entropy Part-of-Speech Tagger [12] and a Brazilian Portuguese language resource [13].

2.2.3. Cross-Domain Filtering

Given that each country has 6 job domains, words that appear across domains in a similar frequency are not domain-specific and should be deleted. This procedure is called cross-domain filtering.

For that, words should first be ranked by their domain-wide TF-IDF weight [14] in each of the domains. Li et al. propose in [15] a formula for computing a corpus-wide TF-IDF weight for a news corpus to get keywords, based on the assumption that keywords appear often in a news article. This is not necessarily the case for job ads where skill requirements may be mentioned only a few times. Wartena et al. present in [16] a corpus-wide TF-IDF weight which term frequency is absolute and therefore gets more importance in the overall result.

Descriptions of ads can be of varying length. Therefore, on top of reducing the weight of term frequency in the TF-IDF formula, we want to reduce the weight of a document's length, such that long job ads will not overshadow shorter ones. We adopt the TF-IDF formula in Equations 1, 2 and 3 introduced by Lee and Kim in [17].

$$TF(w) = \log\left(\frac{1}{|D|}\sum_{d\in D}\frac{n\,(w,d)}{\max_{w'\in d} n\,(w',d)}\right) + 1 \quad (1)$$

$$IDF(w) = \log\left(\frac{|D|}{|\{d \in D : w \in d\}|}\right) \quad (2)$$

$$TFIDF(w) = TF(w) \cdot IDF(w) \quad (3)$$

We perform cross-domain filtering [17] by first ranking the words by their TF-IDF weight, and then we compute the standard deviation of their ranks across the domains. We set a threshold for the standard deviation, here defined as $min(1000, |w \in D|)$ with D being the job ads in the 6 domains for a given country. If the standard deviation is below that threshold, the word w is a word that does not bear domain-specific information and should therefore be deleted.

[2]The lemmatiser was developed at the University of São Paulo by Erick G. Maziero, and is available `here`.

3. Trend Detection

To detect trends, two collections are created for each domain and for each country: one for the first half of 2016, and the other for the first half of 2017.

3.1. Vocabulary Improvement and Filtering

We first form n-grams to collect multi-word phrases and improve the vocabulary, and then we further restrict it to only meaningful, information-rich words by filtering based on named entities and part-of-speech tags.

3.1.1. Forming n-grams

Some skills or other phrases of interest are multi-word expressions. To detect them, we need to form n-grams.

Lent et al. propose in [18] an adaptation of sequential patterns mining to textual data. Agrawal and Srikant introduced the Apriori algorithm in [19] for mining sequential patterns and later improve it in [20] by introducing the Generalised Sequential Patterns (GSP) algorithm.

We use the GSP algorithm to collect all n-grams with $n > 1$ with a minimum support of 0.2. We apply the algorithm on the union of the two collections of job ads compared, so they can share the same n-grams for better trend detection.

3.1.2. Named Entity Filtering

Named entities such as locations and persons should not come up as trends. Indeed, finding that *London* is trending for job ads in the UK is not useful. However, if we find that companies, which are tagged as organisations, are trending, it could be an interesting piece of information.

To perform Named-Entity Recognition (NER) in the job ads in their respective languages, we use Polyglot-NER [21]. We then delete each word or multi-word expression tagged as location or person.

3.1.3. Part-of-Speech Filtering

Verbs and adverbs are examples of parts of speech that will not be meaningful as trends. We therefore decide only to keep words that are tagged as nouns or adjectives or which part of speech could not be determined. The latter case would surface for instance for n-grams or newly coined words, that we would like to keep.

We first detect for each domain of each country the dominant language using automatic language detection in the Polyglot package[3]. Then we associate it with the correct language for part-of-speech tagging with the TreeTagger [22, 23].

3.2. Trend-detecting method

In this subsection, we give statistics on the cases in which the trend-detecting method could be applied, before explaining how TF-IDF weights were used to detect trending words. We also clustered the resulting trending words in a dendrogram using their TF-IDF vectors.

3.2.1. Scope of Application

In the job ads that we collected, not all domains in all countries had data from as far back as the first semester of 2016. Figures are given for the cases in which data was available in Table 1.

[3]The Polyglot package by Rami Al-Rfou is available `here`

Country	Domain	First semester of 2016		First semester of 2017		Common
		Vocabulary Size	Job Ads	Vocabulary Size	Job Ads	Vocabulary Size
Australia	Accounting and Finance	95	36	6,297	7,382	69
	Engineering	111	35	4,379	3,908	86
Brazil	Engineering	424	422	5,334	5,906	228
France	Engineering	670	130	9,699	14,145	459
Germany	Consultancy	161	1,211	10,826	28,533	101
	Engineering	1,208	1,027	23,243	38,647	804
	Information Technology	1,284	1,900	28,133	68,715	1,003
India	Consultancy	71	94	1,620	7,038	15
Netherlands	Accounting and Finance	936	543	9,905	7,798	690
	Engineering	898	445	24,405	37,975	716
Poland	Accounting and Finance	77	1,548	23,006	15,080	33
	Information Technology	222	2,117	18,375	14,106	59
	P.R., Advertising, Marketing	247	125	12,344	3,359	151
Russia	Accounting and Finance	735	497	16,816	85,223	548
	Consultancy	59	193	8,912	24,115	33
	Engineering	166	77	2,287	5,467	68
	Information Technology	297	225	27,466	73,459	254
	P.R., Advertising, Marketing	45	373	17,079	106,809	28
South Africa	Accounting and Finance	150	309	3,385	7,140	97
	Engineering	244	831	1,776	3,884	145
	Information Technology	552	672	7,317	9,135	425
UK	Consultancy	217	31	5,339	21,081	138
	Engineering	640	946	20,244	61,890	402
	Information Technology	301	163	15,127	74,123	217

Table 1: Statistics on cases for which job ads were available from as far back as the first semester of 2016

3.2.2. Comparison of TF-IDF weights

We first compute the TF-IDF weights according to the formula in Equations 1, 2 and 3 for each word separately on the two collections. That formula normalises the term frequency by the maximum term frequency in each individual job ad and by the size of the collection, hereby tackling the imbalance in the number of job ads between 2016 and 2017.

For each word present in both collections, we compute the trend score by subtracting the TF-IDF weight for the first semester of 2016 from its 2017 counterpart. A positive trend score indicates a word that is more trending in the first semester of 2017 than in the same period in 2016.

3.2.3. Clustering of Trending Words

After obtaining the trending words, it is possible to cluster them to get an idea of their similarity.

Wartena and Brussee cluster documents in [24] by using the similarity between their keywords. They expressed that similarity in three ways: the cosine similarity of their document distributions, the Jensen-Shannon divergence of their document distributions and the cosine similarity of their TF-IDF vectors. The latter is widely used in information retrieval and "*has proven to be a robust metric for scoring the similarity between two strings*" [25].

We cluster trending words in a dendrogram with the Unweighted Pair Group Method with Arithmetic Mean (UPGMA) [26]. The distance between two trending words is the euclidean distance between their TF-IDF vectors of length $|D|$ where D is the collection of documents. The distance between two clusters C_1 and C_2 of trending words is the average of all the pairwise distances of the words they contain, as shown in Equation 4.

$$d(C_1, C_2) = \frac{1}{|C_1| * |C_2|} \sum_{a \in C_1} \sum_{b \in C_2} \sqrt{\sum_{d=1}^{|D|} (a_d - b_d)^2} \quad (4)$$

4. Trend Results and Discussion

Based on the pipeline afore described, we obtained trending words for each of the cases in Table 1. We will illustrate the results by focusing on the UK's Information Technology job ads.

We first present the trending words, then we compare the trends obtained to Google Trends in the UK.

4.1. Trending Words

Out of the 217 common words between the 2016 and 2017 collections, 7.8% (17) have a positive difference between their 2017 TF-IDF score and their 2016 one, and are therefore considered to be trending words. They are shown in Figure 1.

The trending words contain skills such as programming languages (Python, PHP, Java), abbreviations (AMP, which stands for Apache, MySQL and PHP, a commonly used solution stack; TDD, Test-Driven Development) or other concepts in Computer Science (Cloud, analytics, troubleshoot).

Figure 2 shows the clustering of the trending words in a dendrogram. This TF-IDF-based clustering captured a few word couples that have similar frequency patterns, like configuration and troubleshoot, or TDD and jQuery.

There is a large part of the words collected that are not trending, although the TF-IDF formula normalised term frequency for each text, and as well for the length of the corpus. It is likely caused by the imbalance in the number of job ads

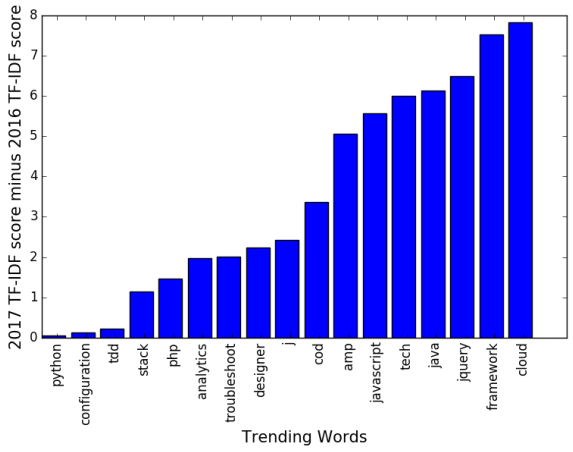

Figure 1: Trending Words in Information Technology jobs in the UK plotted with their 2016-2017 TF-IDF evolution

between 2016 and 2017.

The trending words for the other cases, as well as the dendrograms, are available for public consultation[4].

4.2. Distribution of Trending Words within Job Ad Texts

There were statistical studies about locations of words in corpora. Margulis [27] found in experiments on large documents collections that the assumption that "*the frequency of occurrence of text tokens within the full text documents in a document collection can be described by a sum of Poisson distributions*" is valid for over 70% of frequently occurring terms. Lee and Baik [28] define a keyword in a document collection as a word with high TF-IDF score and high standard deviation in the distribution of its locations within texts.

In our case, a word's position within its text did not influence the possibility that it will become a trending word. We computed the number of occurrences of trending words in the 2017 UK Information Technology job ads, and found the distribution shown in Figure 3. Figure 4 shows a smoothed curve of that distribution. In the 2017 UK IT job ads, the average text length was 112.6, whereas the standard deviation is 129.2. Therefore for the average job ad, the distribution is skewed towards the left, meaning that a trending word is likely to appear at the beginning of a job ad.

We noticed the same results for the other 19 cases where trending words are available[5]. We therefore test whether it is possible to predict the distribution of the position of trending words within the job ads of a certain case knowing the distributions in the other cases. We fit a Generalized Additive Model (GAM) [29, 30] for each case in a leave-one-out fashion and then compute the Normalised Root Mean Square Error (NRMSE). The results obtained are in Figure 5. The prediction is very accurate, as the average NRMSE is only 3.3%, with a median of 2.0% and a standard deviation of 3.5%. It shows a close correlation, regardless of country, language or job domain, between the location of a word in a job ad and its trendiness.

[4]Trending words are available `here`.

[5]Text positions of trending words are available `here`.

4.3. Comparison with Google Trends

Google Trends has been used for trend predictions in other fields. In economics, Carrière-Swallow and Labbé [31] find that Google Trends data improves the prediction of current state (*nowcasting*) of automobile sales in Chile by up to 14%. Choi and Varian [32] build simple seasonal auto-regressive models with relevant Google Trends data for cases including consumer confidence and travel destination planning and find that they outperform similar models without the Google Trends data by 5 to 20 percent. In healthcare, Google Flu Trends can take advantage of the fact that millions of users worldwide will look for information about a disease they worry about and therefore it can detect regional outbreaks of influenza up to 10 days before convential surveillance systems for disease control and prevention [33].

But does it have the same trend prediction power in the job market? For each case with trending words, we query Google Trends for a given trending word w for the corresponding country during the period from January 2016 to June 2017. Each datapoint obtained is an integer representing one month. Google Trends does not give absolute data about search query quantities, but relative data, with the maximum being normalised to 100, and 0 meaning that there was no search query. We compute the trending energy of a word w in a country c and in a set of months M using the formula [3] in Equation 5. It takes into argument the vector v of $|M|$ Google Trends values. A negative value indicates decreasing popularity, and vice-versa for a positive value.

$$energy(v) = \sum_{m=1}^{|M|-1} \left({v_{|M|}}^2 - {v_m}^2 \right) * \frac{1}{|M| - i} \qquad (5)$$

The results for the trending words in the UK's IT jobs are in Figure 6. We notice that the Google Trends energies are conflicting with our results. It is worth noting that our results are definitively domain-specific, whereas word sense disambiguation cannot be done in a google search query for polysemic trending words, such as python, stack, amp and framework. Moreover, whereas the trend energies obtained with Google Trends are representative of queries done by a general audience, our trends stem directly from, and thus give insights of, the IT job market demand, and more broadly domain-specific job market demands.

5. Conclusions

In this paper, we proposed a method to automatically detect trends in job advertisements.

First, we collected job advertisements from 16 countries and in 6 job domains. These job ads were in 8 languages. We pre-processed them in their own language by removing the stop words, lemmatising, and performing cross-domain filtering [17] to remove words that do not bear domain-specific information.

Then, we improve the vocabulary by forming n-grams using the GSP algorithm [20, 19] and further restrict it by filtering based on Named-Entity Recognition [21] and Part-of-Speech Tagging [22, 23]. We then split the job ads to evaluate trends over two periods: the first semester of 2016 and the first semester of 2017. We apply our method on the cases where data was available and compute the TF-IDF weights evolution from 2016 to 2017. A trending word is defined as a word appearing both in 2016 and 2017 and having a TF-IDF weight higher in

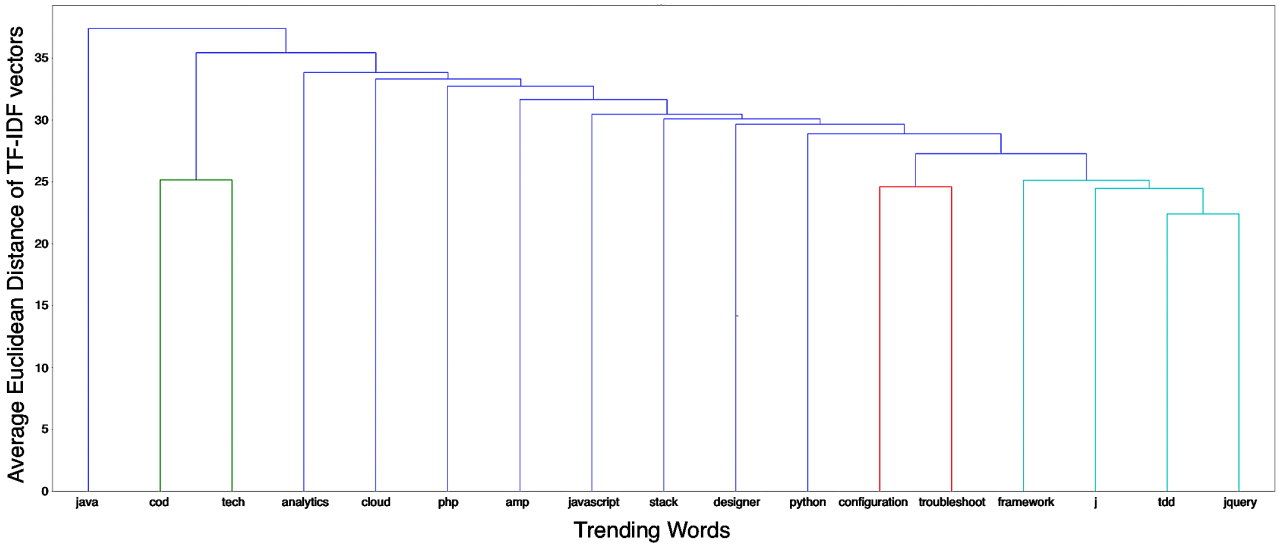

Figure 2: Dendrogram of Trending Words in Information Technology jobs in the UK clustered according to their 2017 TF-IDF vectors

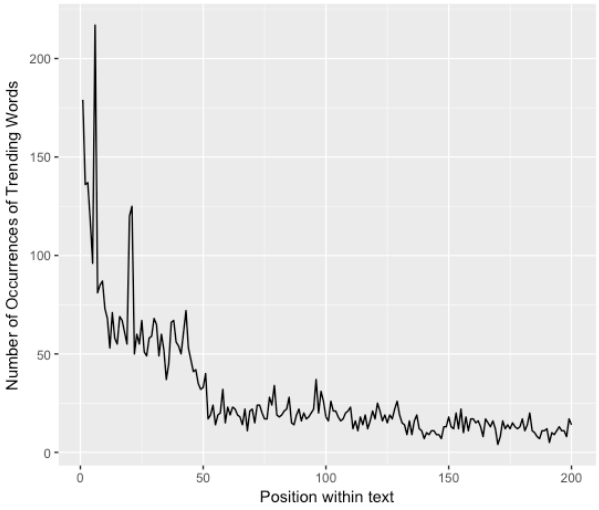

Figure 3: Number of Occurrences of Trending Words in function of Positions within the texts in the 2017 UK Information Technology job ads

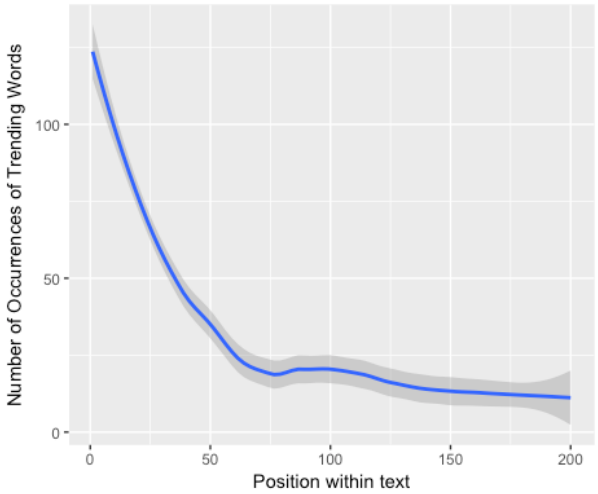

Figure 4: Smoothed Curve of the Number of Occurrences of Trending Words in function of Positions within the texts in the 2017 UK Information Technology job ads

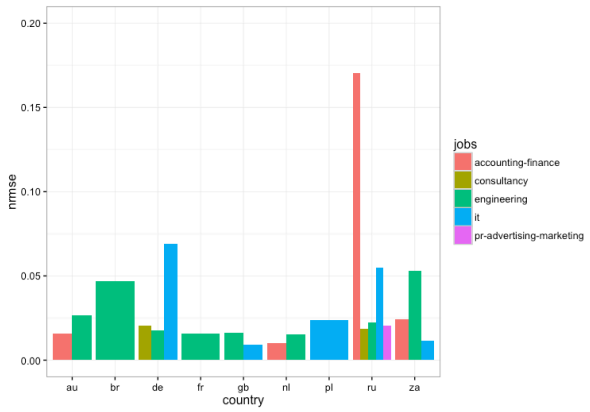

Figure 5: GAM prediction of the distribution of the positions of trending words in cases of job ads plotted in a leave-one-out fashion plotted with their NRMSE

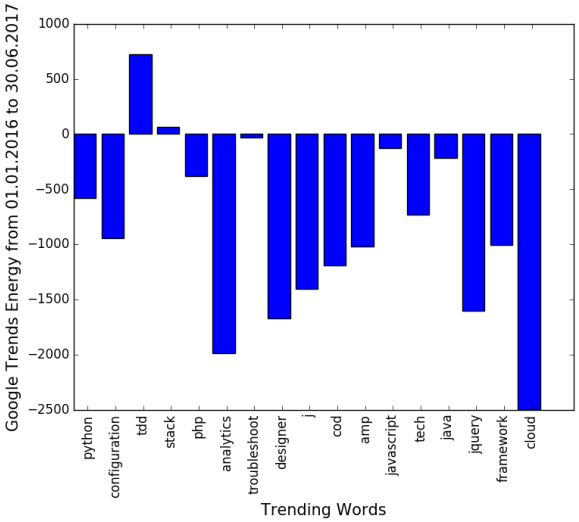

Figure 6: Trending Words in Information Technology jobs in the UK plotted with their Google Trends energy for the time period between 01.01.2016 and 30.06.2017

2017 than in 2016. We cluster keywords in dendrograms using the UPGMA [26].

Finally, we provide the results for all the cases for public consultation, and focus on the UK's Information Technology job ads to illustrate them. The resulting trending words contained skills such as programming languages and Computer Science tools and concepts. The clustering of trending words managed to capture a few meaningful connections.

We then took a look at the distribution of the positions of trending words within job ad texts and found that most of the distributions are skewed to the left, meaning that a trending word is likely to appear at the beginning of job ad texts. We also found that predicting the distribution of a case based on a GAM [29, 30] of the other distributions yields a great accuracy averaging 3.3% NRMSE. This indicates a close correlation of the position of a word in its text and its trendiness, regardless of job domain, language or country.

Knowing Google Trends' efficiency in predictions in economics [31, 32] and disease prevention [33], we test whether the wisdom of the crowd and their search queries give similar results to our method. Google Trends is conflicting with our results, most likely due to the fact that the job ads stem from domain-specific job market demands, rather than a general audience.

6. References

[1] A. Rossett, *Training needs assessment*. Educational Technology, 1987.

[2] J. Brown, "Training needs assessment: A must for developing an effective training program," *Public personnel management*, vol. 31, no. 4, pp. 569–578, 2002.

[3] M. Cataldi, L. Di Caro, and C. Schifanella, "Emerging topic detection on twitter based on temporal and social terms evaluation," in *Proceedings of the tenth international workshop on multimedia data mining*. ACM, 2010, p. 4.

[4] M. Kämpf, E. Tessenow, D. Y. Kenett, and J. W. Kantelhardt, "The detection of emerging trends using wikipedia traffic data and context networks," *PloS one*, vol. 10, no. 12, p. e0141892, 2015.

[5] C. Zhang, "Automatic keyword extraction from documents using conditional random fields," *Journal of Computational Information Systems*, vol. 4, no. 3, pp. 1169–1180, 2008.

[6] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.

[7] S. Bird, "Nltk: the natural language toolkit," in *Proceedings of the COLING/ACL on Interactive presentation sessions*. Association for Computational Linguistics, 2006, pp. 69–72.

[8] C. Fellbaum, Ed., *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[9] T. D. Smedt and W. Daelemans, "Pattern for python," *Journal of Machine Learning Research*, vol. 13, no. Jun, pp. 2063–2067, 2012.

[10] I. Segalovich, "A fast morphological algorithm with unknown word guessing induced by a dictionary for a web search engine." in *MLMTA*, 2003, pp. 273–280.

[11] M. Szymczak, *Słownik języka polskiego*. Wydawn. Nauk. PWN, 1996, vol. 3.

[12] A. Ratnaparkhi *et al.*, "A maximum entropy model for part-of-speech tagging," in *Proceedings of the conference on empirical methods in natural language processing*, vol. 1. Philadelphia, PA, 1996, pp. 133–142.

[13] M. C. Muniz, M. D. G. V. Nunes, and E. Laporte, "Unitex-pb, a set of flexible language resources for brazilian portuguese," in *Workshop on Technology on Information and Human Language (TIL)*, 2005, pp. 2059–2068.

[14] G. Salton and M. J. McGill, "Introduction to modern information retrieval," 1986.

[15] J. Li, Q. Fan, and K. Zhang, "Keyword extraction based on tf/idf for chinese news document," *Wuhan University Journal of Natural Sciences*, vol. 12, no. 5, pp. 917–921, 2007.

[16] C. Wartena, R. Brussee, and W. Slakhorst, "Keyword extraction using word co-occurrence," in *Database and Expert Systems Applications (DEXA), 2010 Workshop on*. IEEE, 2010, pp. 54–58.

[17] S. Lee and H.-j. Kim, "News keyword extraction for topic tracking," in *Networked Computing and Advanced Information Management, 2008. NCM'08. Fourth International Conference on*, vol. 2. IEEE, 2008, pp. 554–559.

[18] B. Lent, R. Agrawal, and R. Srikant, "Discovering trends in text databases." in *KDD*, vol. 97, 1997, pp. 227–230.

[19] R. Agrawal and R. Srikant, "Mining sequential patterns," in *Data Engineering, 1995. Proceedings of the Eleventh International Conference on*. IEEE, 1995, pp. 3–14.

[20] R. Srikant and R. Agrawal, "Mining sequential patterns: Generalizations and performance improvements," *Advances in Database TechnologyEDBT'96*, pp. 1–17, 1996.

[21] R. Al-Rfou, V. Kulkarni, B. Perozzi, and S. Skiena, "Polyglot-NER: Massive multilingual named entity recognition," *Proceedings of the 2015 SIAM International Conference on Data Mining, Vancouver, British Columbia, Canada, April 30 - May 2, 2015*, April 2015.

[22] H. Schmid, "Part-of-speech tagging with neural networks," in *Proceedings of the 15th conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, 1994, pp. 172–176.

[23] ——, "Treetagger| a language independent part-of-speech tagger," *Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart*, vol. 43, p. 28, 1995.

[24] C. Wartena and R. Brussee, "Topic detection by clustering keywords," in *Database and Expert Systems Application, 2008. DEXA'08. 19th International Workshop on*. IEEE, 2008, pp. 54–58.

[25] S. Tata and J. M. Patel, "Estimating the selectivity of tf-idf based cosine similarity predicates," *ACM Sigmod Record*, vol. 36, no. 2, pp. 7–12, 2007.

[26] R. Sokal and C. Michener, "A statistical method for evaluating systematic relationships," *University of Kansas Scientific Bulletin*, vol. 38, pp. 1409–1438, 1958.

[27] E. L. Margulis, "N-poisson document modelling revisited," 1991.

[28] J.-W. Lee and D.-K. Baik, "A model for extracting keywords of document using term frequency and distribution," *Computational Linguistics and Intelligent Text Processing*, pp. 437–440, 2004.

[29] T. Hastie and R. Tibshirani, *Generalized additive models*. Wiley Online Library, 1990.

[30] S. N. Wood, *Generalized additive models: an introduction with R*. CRC press, 2017.

[31] Y. Carrière-Swallow and F. Labbé, "Nowcasting with google trends in an emerging market," *Journal of Forecasting*, vol. 32, no. 4, pp. 289–298, 2013.

[32] H. Choi and H. Varian, "Predicting the present with google trends," *Economic Record*, vol. 88, no. s1, pp. 2–9, 2012.

[33] H. A. Carneiro and E. Mylonakis, "Google trends: a web-based tool for real-time surveillance of disease outbreaks," *Clinical infectious diseases*, vol. 49, no. 10, pp. 1557–1564, 2009.