Temporal Semantic Analysis and Visualisation of Words

Zaikun Xu and Fabio Crestani

Faculty of Informatics
Universitá della Svizzera Italiana (USI)
Lugano, Switzerland
{zaikun.xu,fabio.crestani}@usi.ch

Abstract. Today there are many languages spoken in the world, among which English is the most popular one. However, words in English evolved a lot in history such that it is very difficult for contemporary people to read ancient English articles. There are many changes, such as the mutation of word itself, the migration of word usage from one context to another, etc. It is thus very interesting to understand the temporal evolution of word's semantic across a long span of time. In this paper we look at two datasets: the New York Times and the National Geographic to study the temporal evolution of words. For this purpose a model that can embed word into vectors is needed. Word2Vec is such an neural network model that learns a vector representation for each word in a way that similar words are also similar in the vector space. By similar, I mean that words tends to co-occur in the same context. So, to obtain a temporal Word2Vec representation, a temporal Word2Vec model needs to be trained sequentially, training one individual Word2Vec model for a given dataset in each time period. The temporal Word2Vec model allows us to explore different visualisation techniques of word semantic evolution. Temporal Word cloud, Heatmap, t-distributed stochastic neighbour embedding are some of the techniques that makes the visualisation possible.

1 Introduction

Language understanding is a research issue that has been investigated for centuries. We are specifically interested in the dynamic nature of a language, especially the temporal semantic analysis of English words. A word might change its meaning over time, it might even disappear and a new word substitute it. For example, 'car' is a new word that describes a new transportation tool in the 20th century. The notion of car is evolving since it was introduced. With the advancement of automatic driving technology, the integration of smart embedded systems, the concept of car during the 21th century will be fundamentally different from what it was in the past. In this sense, the evolution of word semantic is a reflection of the evolution of human history. Thus it is really interesting to explore the dynamics of word evolution for the assessment of the dynamics of words, people and events. What's more, for researchers to fully understand

a language, its dynamic evolving nature should be considered. To address such kind of questions, temporal semantic analysis and visualisation is the essential way to uncover mysteries of word semantics.

There are many challenges related to temporal semantic analysis of large chunks of data. Firstly, One need to collect a dataset that spans a long time period such that word semantic meaning evolves. Also, data collected needs to be preprocessed before feeding into a model. Secondly, words need to be represented as numbers so that computers can understand and process them. Lastly, how to visualise temporal semantic of a word is an open question too.

2 Related Work

There are various models developed along the way for language understanding. Bag of words (BoW) [3] is a naïve model to represent a document or a sentence using a n-dimensional vector such that the frequency of each word is represented in each position of a n-dimensional vector, where n is the number of total unique words. The BoW model essentially employs one-hot representation. The problem of one-hot representation is that it linearly scales with the number of words and, more importantly, it can not capture the inherent similarity between two words. For example, the word 'car' and 'truck' are semantically similar, but in one-hot representation, 'car' can be as similar to any other word as 'truck'. On the other side, distributed representation [4], which is to embed each word into a n-dimensional vector space, can directly compare word similarity in the n-dimensional vector space. The Neural Probabilistic Language Model (NPLM), proposed by [2], is a neural network model that utilize distributed representation, which has a input layer, a projection layer, a hidden layer and an output layer. The input layer takes each word and projects it into a n-dimensional vector in the projection layer through a shared matrix C. The embedding vector is then passed from the projection layer to the hidden layer and the embedding is learned through training with Back-propagation (BP) algorithm. The objective function is to maximise the log likelihood of training data. Comparing it with the n-gram model, NPLM can model words with longer distance and the number of parameters scales only linearly with number of unique words, while the n-gram model's computational complexity increases exponentially with the size of unique words. Despite its effectiveness, when the input size is big, the number of neurons in hidden layer and the output size has to be very large to capture the underlying complexity of input data, which makes it computational inefficient and have to be greatly parallelised. Word2Vec [6] was proposed by Mikolov in 2013 for the language modelling of the 6 billion tokens from Google News corpus. As its name suggests, Word2Vec is a model that maps a word into a vector, which is called embedding. More precisely, it is a neural network that is trained on a large corpus of sentences to learn word embeddings such that similar words occur in similar contexts. There are two popular neural network architectures of Word2Vec, one is CBOW (continuous BoW model) and the other is skip-gram , both of which are simple one-hidden layer neural networks. This greatly reduces

the network complexity and makes it computational feasible. Recently, there is an interesting work by [5] that trains a Word2Vec sequentially for temporal time periods to learn a temporal word representation such that words at different time points are different.

3 Data Processing

3.1 Data Collection

Temporal word semantic analysis requires that we have access to a reasonable amount of texts that spans a long period of time. In this era of big data, digitised texts are much easier to access. Many old documents or books are digitised, scanned and uploaded to the Internet. More recent articles or books can be completely digitised at the very time they are created, into different formats. In this study, we choose the National Geographic (NG) and the New York Times (NT) magazine, both of which have digitised texts that spans more than 100 years, facilitating our analysis at a large scale and long period of time. However, the format of data available to us is non-uniform for NT, namely, their articles published before 1922 are in scanned PDF formats while data after 1970s are in html format. Data in between those two period are not available for public download. For the scanned PDF, there are huge variations of fonts, writing styles and scan qualities, imposing great pressure on OCR technologies to transform such PDF documents into texts. Such transformation usually comes in low quality using open-source packages, like for example Tesseract. Besides, there are more than 3 millions scanned PDF, which will takes more than a month for a single computer to process. Due to the lower quality of OCR and intensive computation, we decide to avoid using data before 1922 for NT. Scrapy, an open-source library for data crawling is applied and customised to crawl html data from the NT web site. NG, on the other hands, spans about 110 years, with all articles digitised into 6 DVDs, which can be transformed into text with relative ease and good quality for pure text images. Still, many errors can occur due to the non-uniform layout of images. Figure 1 shows that NT has much more articles each year than those in NG shows in a decade. However, NT spans only 47 years (since 1973) while NG spans more than 100 years.

3.2 Data Normalisation

After data is collected, we corrected errors and applied text normalisation techniques on the raw datasets. Figure 2 shows the text normalisation pipeline. The first step is concerned with common misspelling substitution. Common mistakes exists after OCR processing. For example, the character 'w' in the original PDF is mistakenly processed as 'vv'. This kind of error is caused by the fact that the quality of the input is low and the OCR engine is confused. It is then followed by broken words concatenation. In the OCR processed text, it is very common that one word is broken into two parts especially at the end of a line. Since this

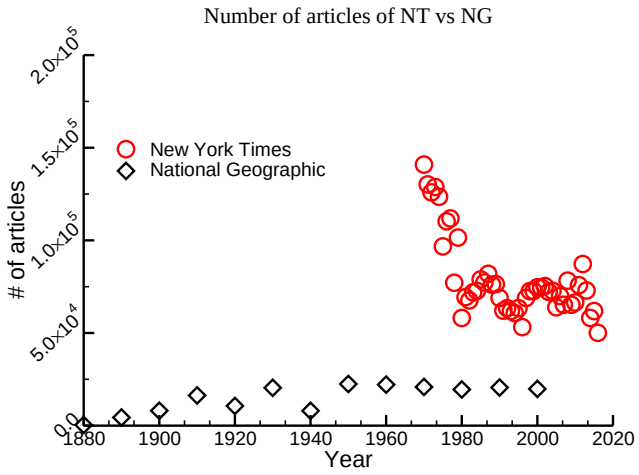

Fig. 1. Data statistics for NT and NG.

occurred a lot, we performed a thorough checking to decide whether concatenation is needed or not for each line of texts. Afterwards, non English words were deleted and stop-words removed. Stemming is not applied in this work since we want to keep the original words.

4 Temporal Analysis

Temporal analysis is a broad topic that try to analyse any data with temporal structure in it, either financial time series data or large collections of text spanning for long period of time. The importance of temporal analysis lies in that it views data as a dynamic evolving structure instead of a static one. So its goal is to find out temporal patterns in the data studied, which are not unveiled by methods overlooking the temporal dimension. In this work, we setup a simple framework by training a temporal Word2Vec model on NG and NT dataset and analysing word's temporal semantic dynamics with respect to an anchored word.

4.1 Problem Definition

The goal of word embedding is to embed words into a vector space such that the similarities between words can be directly measured by vector operations, such as cosine similarities in the corresponding vector space. Word2Vec, is such a powerful model that captures semantic similarity between words co-occurring in similar contexts. In the temporal Word2Vec case, the goal is then to embed words into a discrete vector space with extra temporal dimension. Here we denote V_w as

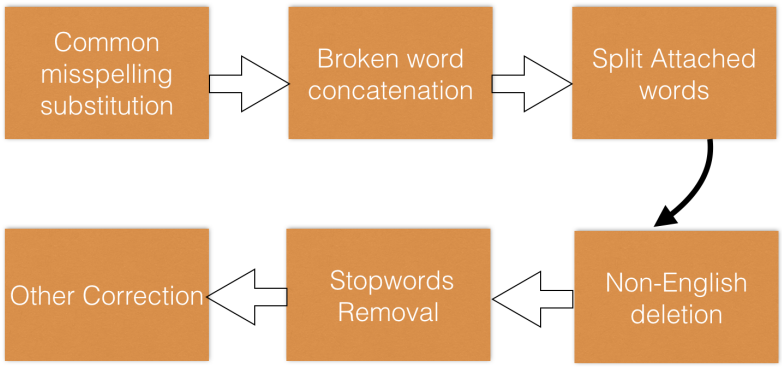

Fig. 2. The Text Normalisation Pipeline.

a temporal vector representing word w, with V_{w,t_i} being the vector representation of word w at time t_i:

$$V_w = \{V_{w,t_1}, V_{w,t_2}, V_{w,t_3}, ..., V_{w,T}\}$$

Ideally, we want the temporal Word2Vec model to capture the evolution dynamics of word semantics such that V_{w,t_i} represents the word w at time t_i. However, the obstacle to obtain such a model is that it is difficult to evaluate a temporal Word2Vec model quantitatively. After all, there is no ground truth of a word semantic representation. So, we focussed this work on the visualisation of the temporal similarity of words and on the evolution of such similarity over time (see next section).

We apply the common way of training a temporal Word2Vec model proposed by [5]. Briefly, the strategy is to train a sequence of Word2Vec model for each time period one by one and each model's weight is initialised based on the one trained before. Then the problem of calculating word similarity for two words w_1,w_2 becomes the following:

$$sim(w_1, w_2) = [sim(w_{1,t_1}, w_{2,t_1}), sim(w_{1,t_2}, w_{2,t_2}), ..., sim(w_{1,T}, w_{2,T})]$$

where w_{1,t_i} means the word w_1 at time t_i and $sim(w_1, w_2)$ is defined as :

$$sim(w_1, w_2) = \frac{w_1 * w_2}{|w_1| * |w_2|}$$

where the L2-norm is used.

4.2 Model Training

For training the temporal Word2Vec model, we use Gensim [7], a Word2Vec library written in Python with Cython optimisation that achieves 70 speedup

comparing it to a plain Numpy implementation. The two dataset are prepared by putting all docs into a text file within one time period where each doc takes one line in the file, so there is a total of T files, where T is the number of time period. For both datasets we trained the temporal Word2Vec model for each time period and use this trained model to initialise the next Word2Vec model. As for hyper parameters, the window size is set to 5 and the embedding dimension is set to 200. After the training finished, a model M_i is saved for each t_i and the vector representation for each word is saved in M_i.

Training the temporal Word2Vec model of NG is fast due to the relative small size of the dataset. While training the temporal Word2Vec model of NT takes much longer time (around 2 hours). The training was done on a Macbook Pro with 16 GB 1600 MHz DDR3 and a 2.2 GHz Intel Core i7 processor.

5 Visual Temporal Analysis

In this section, we visualise the word semantic evolution by exploring different visualisation techniques and adapting those techniques to temporal analysis. Three techniques, Word Cloud, Heatmap and t-SNE are adopted to visualise word vectors from trained Word2Vec model. The example of the anchor word 'car' is show for all the three cases.

5.1 Word Cloud

A Word Cloud visualise the relative frequency of each word by making the more frequent word bigger and the less frequent word smaller. For its layout, words can be positioned horizontal, vertical or oblique and the Word Cloud generator produces the layout of words such that there is no overlap following their size and position constraints. Since we are interested to study the temporal word semantics evolution, the question is whether Word Cloud could be adapted to visualise the evolution of word semantics.

Originally, Word Cloud is based only on one metric, which is the frequency of given words. In order to study word similarities with Word Cloud, the simple change is to adopt the similarity of a word w.r.t. an anchor word as the metric. Consequently, the more similar one word towards the anchor, the bigger the size of that word. We call this approach Semantic Similarity Word Cloud (SSWC).

With SSWCs generated for a certain time period, a possible way to have a Temporal Semantic Similarity Word Cloud (TSSWC) is to generate SSWC in each time period and concatenate them sequentially into a GIF. From a visual point of view it would be nice to have a word position fixed in each SSWC to facilitate eye tracking. The algorithm 1 below shows the pseudo-code that calculates the font size of each word proportionally to its similarity weight. Figure 3 shows the SSWC plot of NG for two time periods, namely, the first decade in 20th and 21th century. Of course we cannot show in this paper a video of the entire sequence of years. Also, due to space constraint, the exact location of the

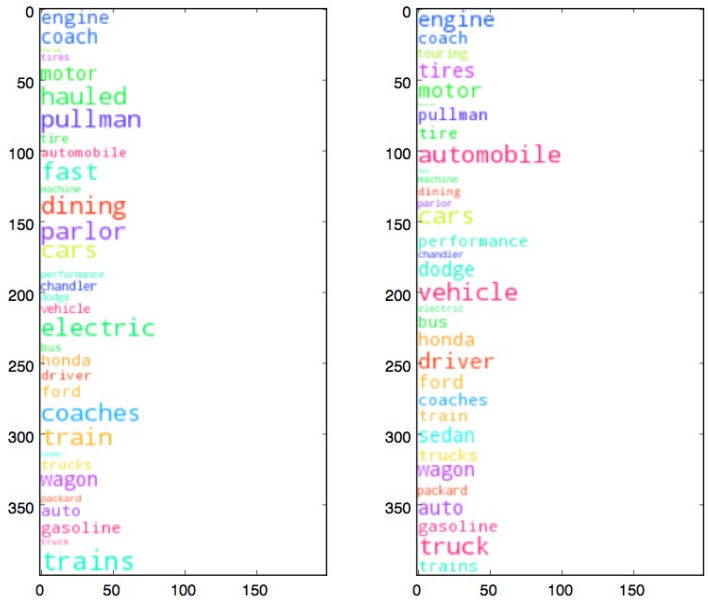

Fig. 3. SSWC for 1900s (left) and 2000s (right) with the words most similar to 'car' according to the National Geographic.

same word in two SSWC is still not exactly the same since each word size can be different.

Note that not all SSWCs are shown in Figure 3, rather only two representative ones are shown here. As you can see, the relative order of each word in each SSWC plot is fixed. Immediately one can pick up some difference for the same word between 1900s and 2000s. In the 1900s, 'parlor' is more similar to 'car' as well as 'pullman'. However, in the 2000s, the notion of 'car' is less similar to 'parlor' and more similar to 'automobile'. In fact, the word 'pullman' in the USA, was specifically applied to refer to railroad sleep cars which were operated by the Pullman Company from 1867 to 1968. Thus, in 2000s, the word 'pullman' should not be similar to car anymore due to its disuse in car related contexts. The word 'automobile' is derived from an Ancient Greek word, meaning 'self'. Over time, the word 'automobile' is less used in Britain but remains widely used in North America.

Figure 4 shows the same SSWC plot for the NT dataset. The two series of words between NT and NG are not exactly the same since contexts are different and a word that is similar to 'car' in NG is not necessarily similar according to th

NT. Overall, the notation of car is kind of fixed since 1970s, such as four-wheeled, powered by gasoline, famous brands including Ford, Toyota, BMW.

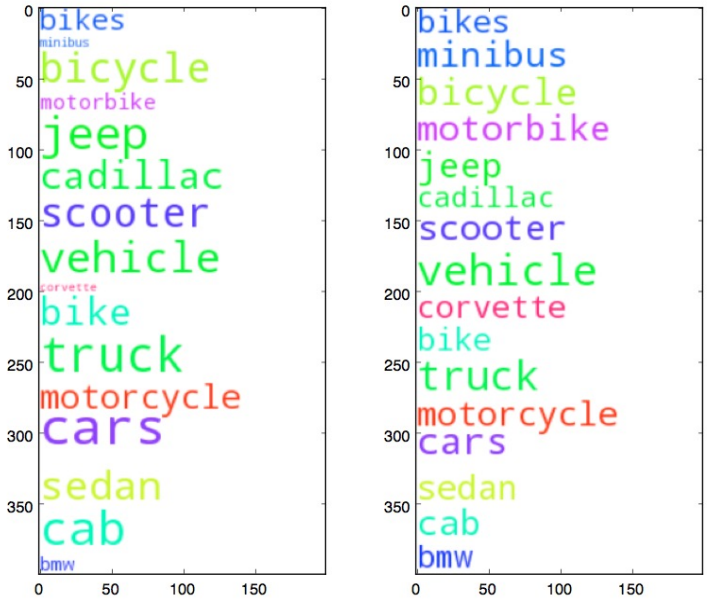

Fig. 4. SSWC for 1970 (left) and 2016 (right) with the words most similar to 'car' according to the New York Times.

5.2 Heatmap

In this section, we explore another visualisation technique, Heatmap. The idea of an Heatmap is that each value is associated with a color. The task is always to visualise word similarities for each temporal period. In the temporal word semantic analysis case we let y-axis be a list of each words and x-axis be a list of each temporal periods such that for a given word we can get an understanding of its temporal similarity to the anchor word quickly based on the change of color.

Figure 5 shows the Heatmap visualisation of words that are most similar to the word 'car' for both NG and NT. Since for each time period, the most similar words to a given word is different, we choose the union of the top 10 most similar words for each time period and list them along the y-axis. Note that word similarity is normalised across all its time period for a better contrast of colours. So:

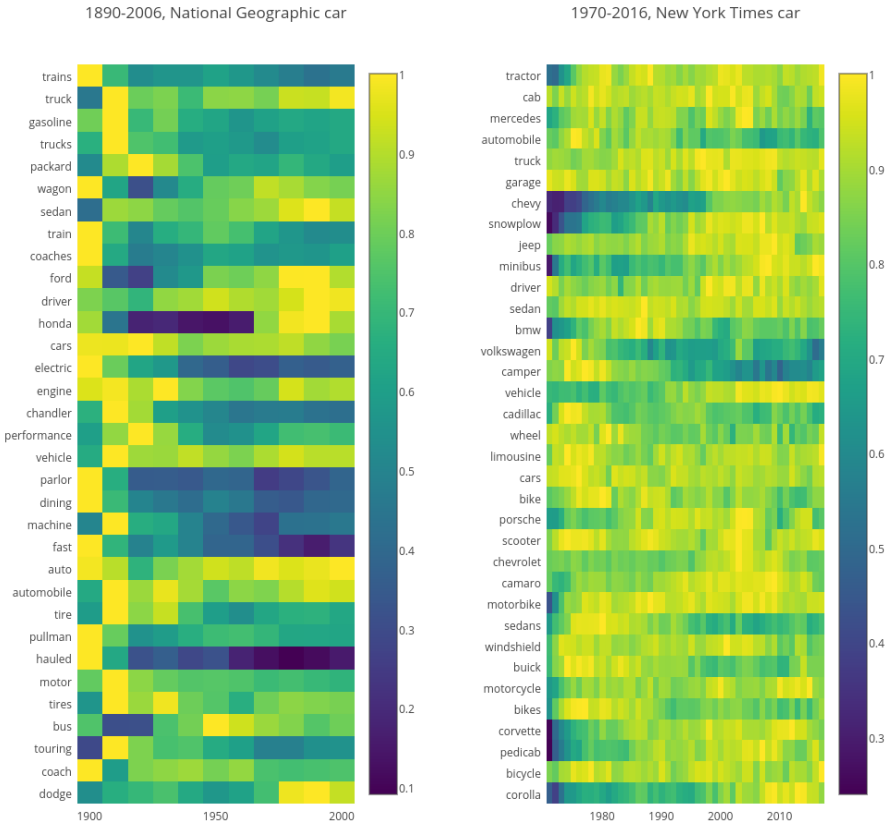

Fig. 5. Heatmap visualisation of word similar to the word 'car'.

$$sim_{w_{j,tt},w_{ref}} = \frac{sim_{w_{j,tt},w_{ref}}}{\sum_{t=1}^{n} sim_{w_{j,t},w_{ref}}}$$

where n is the length of the most similar word list, $w_{j,tt}$, w_{ref} is the j-th word at time period tt and anchor word respectively.

One interesting phenomenon is that in figure 5 right has more car brands than figure 5 left. Actually, there are 9 brands in figure 5 right, which are mercedes, bmw, jeep, chevy, volkswagen, cadillac, camaro, buick, corolla. While in the figure 5 left, only ford, honda, chandler dodge are mentioned. One way to interpret this phenomenon is that globalisation lead to many more international cars being introduced into the US market in 2016 than in 1970.

Now, let's look at each individual case in figure 5 left. For example, the word 'chandler' is very similar to car before 1930s and less similar afterwards. Interestingly, when checking the history we find that the Chandler Motor Car,

a company founded in 1913 whose production peaked in 1927, purchased by a competitor two years later. The history of the Chandler Motor Car is roughly synchronised with the dynamics of the semantic similarity between the word 'Chandler' and the word 'car'.

Another interesting word is 'honda'. The brand Honda became an important motorcycle and car provider in America. It went on to dominate the America's car and motorcycle market with as high a percent marketshare as 63% in 1966 starting from 0% in 1959. This history is reflected by the Heatmap colours.

Interestingly, the word 'ford' is not related to the word 'car' during the early decades of 20th century, despite the well-known Ford's T-car that dominated the market at that time. One way to explain this inconsistency is to refer to the fact that 'car' was more similar to 'train' at that time, showing that there might be a delay between a word semantic meaning and the true concept of a word.

Another interesting word 'dining', whose similarity peaks in the late 19th century and early 20th century. According to Wikipedia, the concept of 'dining car' can be traced back to 1880s when dining cars were a normal part of long-distance travelling trains. This also explains the phenomenon that word 'train' have roughly the same pattern of similarity as 'dining' to the word 'car'.

6 Conclusions

In this paper we studied the temporal semantic evolution of words in two datasets: the National Geographic and the New York Times. The National Geographic spans more than 100 years but have less articles per day and covers less topics. The New York Times (available to us), on the other hand, covers only 47 years since 1970, but have much more articles and has a much more broader topic coverage. They are both America magazines; NG focuses more on geography, history and cultures, while NT focuses on politics, life, society and opinions.

We applied Word2Vec model, which is a neural network language model that maps each individual word into a vector space where vectors can be measured by cosine-similarity. However, the Word2Vec model assumes that one word does not change over time, which is not true. In order to take word semantic evolution into account, we built a temporal Word2Vec model where each word is mapped into a vector of vectors, namely, one vector for each time period t. There are other ways to train such a temporal Word2Vec model in addition to the sequential initialisation procedure. For example, Alburg et al [1] initialise all the models with the same pre-trained weights and adds a regularisation term to the model.

Two visualisation techniques were explored: Word Clouds and Heatmaps. Word Cloud visualise words based on their frequency. TSSWC is adapted from Word Cloud by fixing each word's position and horizontal orientation. Although the Word2Vec approach has advantages in temporal word analysis, it also has certain limitations. First, some words do not exist in the early years, which leads to the exclusion of such word in the word set of Word2Vec model. For example, the word 'Internet' is not mentioned in the 70s which makes it impossible to compare the similarity between the word Internet' and 'surfing', for example.

Also, 'computer' does not exists in late 19th century in the NG dataset. Accordingly, the temporal Word2Vec framework does not have the flexibility to detect when a new word occurs and then show its dynamics from that period. Secondly, one word is a basic unit for constructing a sentence in English. However, this is not always true. For example, human names are composed by at least two words, family name and name.

Nonetheless, the visualisation analysis in this paper is effective and fruitful. It captures many interesting trends of word semantic similarities. What is interesting is to explore the possible reasons leading to such trends. Overall, temporal Word2Vec model and the other visualisation techniques introduced here provide researchers a great combination of tools to spot word semantic evolution, despite the presence of some noise.

Acknowledgments.

This work was extracted from the first author's Master Thesis submitted at the Faculty of Informatics of the Universitá della Svizzera Italiana (USI) in January 2017. The work was supervised by the second author.

References

1. Alburg, H.: Tracking temporal evolution in word meaning with distributed word representation (2015), master thesis in Computer Science, Chalmers University of Technology
2. Bengio, Y., Ducharme, R., Vincent, P., Janvin, C.: A neural probabilistic language model. J. Mach. Learn. Res. 3, 1137–1155 (Mar 2003)
3. Harris, Z.: Word. Distributed Structure (1954)
4. Hinton, G.E.: Learning distributed representations of concepts. Proceedings of the eighth annual conference of the cognitive science society (1986)
5. Kim, Y., Chiu, Y.I., Hanaki, K., Hegde, D., Petrov, S.: Temporal Analysis of Language through Neural Language Models. ArXiv e-prints (May 2014)
6. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. CoRR abs/1301.3781 (2013)
7. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. pp. 45–50. ELRA, Valletta, Malta (May 2010)