Retrieving Word Associations with a Simple Neighborhood Algorithm in a Graph-Based Resource

Gemma Bel-Enguix
LIF
Aix Marseille Université,
13288 Marseille
gemma.belenguix@gmail.com

Abstract

The paper explains the procedure to obtain word associations starting from a graph that has not been specifically built for that purpose. Our goal is being able to simulate human word associations by using the simplest possible methods, including the basic tools of a co-occurrence network from a non-annotated corpus, and a very simple search algorithm based on neighborhood. The method has been tested in the Cogalex shared task, revealing the difficulty of achieving word associations without semantic annotation.

1 Introduction

Building annotated computational resources for natural language is a difficult and time-consuming task that not always produces the desired results. A good alternative to semantic annotation by hand could be using statistics and graph-based operations in corpora. In order to implement a system capable to work with such methods we have designed co-occurrence networks from large existing corpora, like Wikipedia or the British National Corpus (Burnard & Aston, 1998). The underlying idea is that systems based on mathematics and statistics can achieve comparable results to the ones obtained with more sophisticated methods relying on semantic processing.

Non-annotated networks have been suggested and implemented, for example, by Ferrer-i-Cancho and Solé (2001). The authors suggested non-semantically annotated graphs, building exclusively syntagmatic networks. By this method, they reduced the syntagmatic-paradigmatic relations. The authors used the BNC corpus to build two graphs G1 and G2. First, a so-called co-occurrence graph G1 in which words are linked if they co-occur in at least one sentence within a span of maximal three tokens. Then a collocation graph G2 is extracted in which only those links of G1 are retained whose end vertices co-occur more frequent than expected by chance.

A non-annotated graph built from a large corpus (Bel-Enguix and Zock, 2013) is a good representation to allow for the discovery of a large number of word relationships. It can be used for a number of tasks, one of them being computing word associations. To test the consistence of the results obtained by our method, they will be compared with the Edinburgh Association Thesaurus, a collection of 8000 words whose association norms were produced by presenting each of the stimulus words to about 100 subjects each, and by collecting their responses. The subjects were 17 to 22 year old British students. To perform the tests, we take a sample (EAT: http://www.eat.rl.ac.uk/) consisting in 100 words.

For building a network to deal with the specific task of producing word associations we have used the British National Corpus (BNC) as a source.

The way the network has been constructed has also some interest and impact in the final results. Firstly, for the sake of simplicity, we removed all words other than Nouns and Adjectives. Nouns have been normalized to singular form. After this pre-processing, a graph has been built where the nouns and adjectives in the corpus are the nodes, and where the edges between these nodes are zero at the beginning, and are incremented by 1 whenever the two respective words co-occur in the corpus as direct neighbors (i.e. more distant neighborhood was not taken into account). That is, after processing the corpus the weight of each edge represents the number of times the respective words (nodes) co-occur.

To build the graph our system runs through a pipeline of four modules:

- document cleaning (deletion of stop-words), extracting only ‘Nouns’ and ‘Adjectives’;
- lemmatisation of word forms to avoid duplicates (horse, horses);
- computation of the (un-directed) graph's edges. Links are created between direct neighbours;
- computation of the edges’ weights. The weight of an edge is equal to the number of its occurrences. We only use absolute values.
- computation of the node’s weights. As in the edges, the weight of a node is the number of it occurrences.

The graph has been implemented with Python.

The resultant network has 427668 nodes (different words). Of them, 1894 are happax (occur only once), only the 0,5%. There are 13654814 edges. From them, 9836987 with weight one; and 3817827 have a weight higher than one, on a percentage relation 72/28. The average degree of the nodes of the network is 31, 92.

2 Searching method

The search of the target word in the graph has two different steps:

1. Determining the set of common neighbors of the clues,
2. Ranking the set of nodes obtained in 1, and picking the ‘best result’.

2.1 Search of neighbors

The search of the target word T in a graph G, is done via some clues $c_1, c_2,\ldots, c_n$, which act as inputs. G=(V, E) stands for the graph, with V expressing the set of vertices (words) and E the set of edges (co-occurrences). The clues $c_1, c_2,\ldots, c_n \in V$. N(i) expresses the neighbourhood of a node i $\in$V, and is defined as 'every j$\in$V | $e_{i,j} \in$E. The search algorithm is as follows:

- Define the neighbourhood of $c_1, c_2,\ldots, c_n$ as $N(c_1), N(c_2),\ldots, N(c_n)$;
- Get the set of nodes $V_T = N(c_1) \cap N(c_2) \cap \ldots \cap N(c_n)$ and consider $V_c=\{c_1, c_2,\ldots, c_n\}$ to be the set of nodes representing the clues. We define a subgraph of G, G_T, that is a complete bipartite graph, where every element of V_T is connected to every element of V_c;

In the Cogalex shared task, five clues have been given, belonging to any grammatical category, and in different inflected forms (ie., am, be, been or horse, horses). Since the graph has the limitation of containing only Nouns and Adjectives, the system dismisses every word not belonging to the set of nodes V and uses only the remaining clues. And being the words lemmatized, inflected forms are reduced to only one. Therefore, the application will never find ‘be’ from ‘am’, ‘been’, ‘is’.

To build the graph and perform the search, a Python module has been used, Networkx (https://networkx.github.io/), that is extremely fast and efficient.

2.2 Ranking the nodes

This task has been designed with a very simple algorithm. Let’s consider C the number of final stimulus words; $wc_1,wc_2,\ldots,wc_n$ is the weight in the graph of every node c $\in V_C$; $wt_1,wt_2,\ldots,wt_n$ the weight in the graph of every one of the nodes t $\in V_T$; we_{tc} the weight for every edge of G_T, where c $\in V_C$ and t$\in V_T$.

The nodes of the graph are gathered in groups in a logarithmic scale: up to 10^1, 10^2, 10^3, 10^4, 10^5, 10^6. We name *a* the power of 10, ie., for 10^6, *a*=6.

The nodes of VT are ranked with a simple algorithm, consisting in calculating W_t for every $t \in V_T$, so as $W_t= \frac{(we_{tc1}+we_{tc2}+\cdots+we_{tcn})/C}{a}$

The nodes are ranked according to the values of W.

3 Results

In some initial tests, the results were compared with the ones obtained in a sample of the Edinburgh Association Thesaurus (EAT: http://www.eat.rl.ac.uk/) consisting in 100 words. The EAT (Kiss et al., 1973) has 8000 words, and the 100 selected for the test were all of them nouns or adjectives, what made the working easier for our system. There were 15 words that match the ones observed in the Edinburgh Associative Thesaurus (EAT) as Primary Response (PR). There is a partial coincidence – the word given has not a 0 in the EAT – in 54 of the outputs. This means that in more than 50% of the cases the method retrieves a word corresponding to the one produced by a human in the association experiment. This does not imply though that it is the most popular one.

Some other methods of evaluation (Evert & Krenn, 2001) have been applied to the system (Bel-Enguix et al., 2014), showing that the outcomes provided by the graph-based method are quite consistent with human responses, and even optimize them in some specific classes.

In contrast with these results, the ones obtained in the Cogalex shared task were clearly worse. From a total of 200 items, the number of matches was 182, which means an accuracy of the 9,10%.

There are several reasons for that: a) some of the targets were not Nouns or Adjectives, what makes them not retrievable for the system, b) many stimulus words were not Nouns or Adjectives, what makes the algorithm weaker, because such words are dismissed as clues, c) stimulus were not lemmatized and the lemmatization process for words without a context is not easy for the python lemmatization module, d) probably many of the words of the first tested sample were very well-known relations, while the ones in the Cogalex shared task could be less well-connected nodes, e) the ranking algorithm can be clearly improved in order to retrieve the best word, not only one in the list, because we have been asked only full matches.

4 Conclusions: strengths and weakness of the method

Even though the results obtained were not good, there are several strengths that make this system worth to be improved in the future.

Firstly, the network is easy to built and program is very fast. We have used the python package ‘networkx’ to build the graph, integrating its commands into the python script. The result is that in less than one minute the system can compute the two thousand associations that were required. Therefore, while an important improvement is needed in the ranking algorithm, there is room for it, because the performance of the method can afford it.

Secondly, the system works with any co-occurrence graph made from any corpus. This allows us to use specialized corpora as a basis, as well as collections of texts closer to the time the human associations have been produced.

However, there are important weaknesses in the procedure. In the first place, it is necessary to use a network resource including other grammatical categories, at least verbs and adverbs. Even though such graph exists, the difficulty in the application of the current ranking algorithm makes it not-usable so far for this specific task. There is still another clear difficulty in the method, related to the one we just stated: the lack of clustering. Not using semantic annotations is one of our axioms, because it makes the system heavier. Nevertheless, a way to detect which words are more related is needed. This is currently the strongest weakness of this graph-based algorithm. We propose for the future a very simple clustering based on WordNet *synsets* (Miller, 1990), in a way the search can be oriented towards the best choices for every word connection, even though their weight in the graph is lower.

5 Aknowledgements

I am very grateful to my colleagues Michael Zock and Reinhard Rapp for their expertise, comments and constant support.

This work has been supported by the European Commission under a Marie Curie Fellowship, project DynNetLAc.

References

Bel-Enguix, G., Rapp, R. and Zock, M. (2014) A Graph-Based Approach for Computing Free Word Associations, Proceedings of LREC 2014, Ninth International Conference on Language Resources and Evaluation, 3027-3033.

Bel-Enguix, G. and Zock, M. (2013). Lexical Access via a Simple Co-occurrence Network, Proceedings of TALN-RECITAL 2013, 596-603.

Burnard, L. and Aston, G. (1998). *The BNC Handbook: Exploring the British National Corpus*. Edinburgh: Edinburgh University Press

Evert, S. and Krenn, B. (2001). Methods for qualitative evaluation of lexical association measures. In *Proceedings of the 39th Annual Meeting of the Association of Computational Linguistics,* Toulouse, France, 188-915.

Ferrer-Cancho, R., Solé, R. (2001). The small-world of human language. *Proceedings of the Royal Society of London. Series B, Biological Sciences* 268 (2001) 2261-2265.

Kiss, G.R., Armstrong, C., Milroy, R., and Piper, J. (1973). An associative thesaurus of English and its computer analysis. In: A. Aitken, R. Beiley, N. Hamilton-Smith (eds.): *The Computer and Literary Studies*. Edinburgh University Press.

Miller, G. (1990). Wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4).