BOOTSTRAPPING WORD ORDER

Raycho Mukelov

PhD student at Technical University of Varna, Bulgaria, raicho.mukelov@tu-varna.bg

Abstract: In this paper is proposed a methodology for bootstrapping word order based on general to specific noun relations extracted from web corpora frequency counts. For the purpose is employed the TextRank algorithm, asymmetric association measures, web search engine word counts and iterative k-means hard clustering. In addition is proposed a reliability evaluation of the method.

Keywords: Asymmetric Association Measures, Graph Ranking Algorithms, Semantic Word Relations, Hypernymy, Hyponymy, Ordering Words, Taxonomy Labeling, Reliability

ПОДРЕЖДАНЕ НА ДУМИ С МИНИМУМ РЕСУРСИ

Райчо Мукелов

Докторант в ТУ - Варна, България, raicho.mukelov@tu-varna.bg

Резюме: В тази статия е представена методология за подреждане на думи с минимум ресурси базирана на общо-специфични релации извлечени от Уеб корпус. За целта се използва алгоритъма TextRank, асиметрични мерки за асоциация, честотите на думите измерени с помощта на Уеб търсачка и итеративен k-means алгоритъм за клъстеризация. Също така се предлага надеждностна оценка на методът.

Ключови думи: Асиметрични мерки за асоциация, Алгоритми за ранкиране базирани на графи, Семантична подобност на думи, Хиперними, Хипоними, Подреждане на думи, Определяне на категории в таксономии, Надеждност

1. INTRODUCTION

Taxonomies are crucial for any knowledge-based system. They are in fact important because they allow us to structure information, thus fostering their search and reuse. However, it is well known that any knowledge-based system suffers from the so-called knowledge acquisition bottleneck, i.e. the difficulty to actually model the domain in question. As stated in [2], WordNet has been an important lexical knowledge base, but it is insufficient for domain specific texts. So, many attempts have been made to automatically produce taxonomies [8], but [2] is certainly the first work which proposes a complete overview of the problem by:

(1) Automatically building a hierarchical structure of nouns based on bottom-up clustering methods and

(2) Labeling the internal nodes of the resulting tree with hypernyms from the nouns clustered underneath by using patterns such as “B is a kind of A.

This paper is dealing with the second problem of the construction of an organized lexical resource i.e. ordering the words according to their general to specific relations, so the correct nouns are chosen to label internal nodes of any hierarchical knowledge base, such as the one proposed in [4]. There are many different types of semantic relations between words and in this paper is presented a method to deal with general-specific noun relations (including hypernymy/hyponymy relations)

and use them to produce order between the words. By doing this, one is able to correctly label internal nodes of taxonomy in an unsupervised fashion.

2. RELATED WORK

Most of the works proposed so far are pattern based and they:

(1) Use predefined patterns to discover hypernym/hyponym relations or

(2) Automatically learn these patterns to identify hypernym/hyponym relationships.

From the first paradigm [9] first identifies a set of lexical-syntactic patterns that are easily recognizable i.e. occur frequently and across text genre boundaries. These can be called seed patterns. Based on these seeds, she proposes a bootstrapping algorithm to semi-automatically acquire new more specific patterns. Similarly, [2] uses predefined patterns such as "*X* is a kind of *Y*" or "*X*, *Y*, and other *Zs*" to identify hypernym/hyponym relationships. This approach to information extraction is based on a technique called *selective concept extraction* as defined by [15]. Selective concept extraction is a form of text skimming that selectively processes relevant text while effectively ignoring surrounding text that is thought to be irrelevant to the domain.

A more challenging task is to automatically learn the relevant patterns for the hypernym/hyponym relationships. In the context of pattern extraction, there are many approaches as summarized in [19]. The most well-known work in this area is certainly the one proposed by [18] who uses machine learning techniques to automatically replace hand-built knowledge. By using dependency path features extracted from parse trees, they introduce a general-purpose formalization and generalization of these patterns. Given a training set of text containing known hypernym pairs, their algorithm automatically extracts useful dependency paths and applies them to new corpora to identify novel pairs. [18] uses a similar way as [17] to derive extraction patterns for hypernym/hyponym relationships by using web search engine counts from pairs of words encountered in WordNet. However, the most interesting work is certainly proposed by [1] who extract patterns in two steps. First, they find lexical relationships between synonym pairs based on snippets counts and apply wildcards to generalize the acquired knowledge. Then, they apply a SVM classifier to determine whether a new pair shows a relation of synonymy or not, based on a feature vector of lexical relationships. This technique could be applied to hypernym/hyponym relationships although the authors did not mention it.

On the one hand, links between words that result from manual or semi-automatic acquisition of relevant predicative or discursive patterns [2], [9] are fine and accurate, but the acquisition of these patterns is a tedious task that requires substantial manual work.

On the other hand, the works done by [1], [17], [18] and [16] have proposed methodologies to automatically acquire these patterns mostly based on supervised learning to leverage manual work. However, training sets still need to be built.

Unlike other approaches here is proposed an unsupervised methodology, which aims at discovering general-specific noun relationships that can be assimilated to hypernym/hyponym relationships detection[1]. The advantages of this approach are clear as it can be applied to many languages and to any domain without any previous knowledge, based on a simple assumption: specific words tend to attract general words with more strength than the opposite. As [11] states: "there is a tendency for a strong forward association from a specific term like *adenocarcinoma* to the more general term *cancer*, whereas the association from *cancer* to *adenocarcinoma* is weak".

[1] Other kinds of relationships may be covered. For that reason, this paper considers general-specific relationships instead of strictly hypernym/hyponym relationships.

Based on this assumption, which is also similar to the work of [21], is proposed a methodology based on directed graphs and the TextRank algorithm [12] to automatically induce general-specific noun relationships from web corpora frequency counts. Indeed, asymmetry in Natural Language Processing can be seen as a possible reason for the degree of generality of terms [11] and therefore can be used for hypernymy/hyponymy detection.

So, different asymmetric association measures can be implemented to build the graphs upon which the TextRank algorithm is applied and produces an ordered list of nouns, from the most general to the most specific.

Experiments have been conducted based on the WordNet noun hierarchy and assessed that 70% of the words are ordered correctly when compared to WordNet. We must also emphasize that in this work WordNet is used for the purpose of testing and validating the proposed hypothesis and this work doesn't rely on WordNet. WordNet is not needed to discover general to specific word relations and is used for evaluation purposes only.

Software reliability and reliability in general is also an important research topic. Reliability is intrinsic to the Electrical Engineering but it can be also applied in the Computer Science and the Software Engineering. In the recent years, some research has been done in the area of the reliability of electronic circuits and components like in [5], [6] and [7]. For the purpose of the reliability evaluation of the proposed methodology is employed a method inspired by the ones described in the former papers.

3. ASYMMETRY BETWEEN WORDS

In [8], the authors clearly point at the importance of asymmetry in Natural Language Processing. In particular, the author deeply believes that asymmetry is a key factor for discovering the degree of generality of terms.

It is cognitively sensible to state that when someone hears about *mango*, he may induce the properties of a *fruit*. However, when hearing *fruit*, more common fruits will be likely to come into mind such as *apple* or *banana*. In this case, there exists an oriented association between *fruit* and *mango* (*mango* → *fruit*) which indicates that *mango* attracts more *fruit* than *fruit* attracts *mango*. As a consequence, *fruit* is more likely to be a more general term than *mango*.

Based on this assumption, asymmetric association measures are necessary to induce these associations. [14] and [20] propose exhaustive lists of association measures from which are presented the asymmetric ones that can be used to measure the degree of attractiveness between two nouns, x and y, where $f(x,y)$, $P(*)$, $P(x,y)$ and N are respectively the frequency function, the marginal probability function, the joint probability function and the total number of bigrams.

$$Braun\text{-}Blanquet = \frac{f(x,y)}{max(\ f(x,y)+f(x,\bar{y}),f(x,y)+f(\bar{x},y))} \tag{1}$$

$$J\ measure = max\begin{bmatrix} P(x,y)log\frac{P(y|x)}{P(y)}+P(x,\bar{y})log\frac{P(\bar{y}|x)}{P(\bar{y})}, \\ P(x,y)log\frac{P(x|y)}{P(x)}+P(\bar{x},y)log\frac{P(\bar{x}|y)}{P(\bar{x})} \end{bmatrix} \tag{2}$$

$$Confidence = max[P(x|y),P(y|x)] \tag{3}$$

$$Laplace = max\left[\frac{N.P(x,y)+1}{N.P(x)+2}, \frac{N.P(x,y)+1}{N.P(y)+2}\right] \quad (4)$$

$$Conviction = max\left[\frac{P(x).P(\bar{y})}{P(x,\bar{y})}, \frac{P(\bar{x}).P(y)}{P(\bar{x},y)}\right] \quad (5)$$

$$Certainty\ Factor = max\left[\frac{P(y|x)-P(y)}{1-P(y)}, \frac{P(x|y)-P(x)}{1-P(x)}\right] \quad (6)$$

$$Added\ Value = max|P(y|x)-P(y), P(x|y)-P(x)| \quad (7)$$

All seven definitions show their asymmetry by evaluating the maximum value between two hypotheses i.e. by evaluating the attraction of *x* upon *y* but also the attraction of *y* upon *x* where *x* and *y* are two words in consideration. As a consequence, the maximum value will decide the direction of the general-specific association i.e. ($x \rightarrow y$) or ($y \rightarrow x$).

All these measures can be computed by using search engine counts. The only problem may be that some of the formulas need the total number of pages indexed by the search engine which unfortunately is a number that is hard to be determined. For the experiments in this work is considered that the total number of pages is 10^{10}.

In fact, due to the nature of those formulas that need the total number of pages it is sufficient if we just choose e number that is big enough.

In the experiments for example there was no difference at all when the total number of pages is considered to be 10^{10} or 20^{10}. Like this we approximate the probabilities in the cases where we need the total number of pages to compute the value of given association measure and for our work it was working good enough.

4. WORD ORDER DISCOVERY WITH THE TEXTRANK ALGORITHM

Graph-based ranking algorithms are essentially a way of deciding the importance of a vertex within a graph, based on global information recursively drawn from the entire graph.

The basic idea implemented by a graph-based ranking model is that of voting or recommendation. When one vertex links to another one, it is basically casting a vote for that other vertex. The higher the number of votes that are cast for a vertex, the higher the importance of the vertex.

Moreover, the importance of the vertex casting the vote determines how important the vote itself is, and this information is also taken into account by the ranking model. Hence, the score associated with a vertex is determined based on the votes that are cast for it, and the score of the vertices casting these votes.

The intuition of using graph-based ranking algorithms is that more general words will be more likely to have incoming associations as they will be associated to many specific words. On the opposite side, specific words will have few incoming associations, as they will not attract general words (see *Figure 1*).

As a consequence, the voting paradigm of graph-based ranking algorithms should give more strength to general words than specific ones, i.e. a higher voting score.

For that purpose, we first need to build a directed graph. Informally, if *x* attracts more *y* than *y*

attracts *x*, we will draw an edge between *x* and *y* as follows (x → y) as we want to give more credits to general words.

Formally, we can define a directed graph $G = (V, E)$ with the set of vertices V (in this case, a set of words) and a set of edges E where E is a subset of $V \times V$ (in this case, defined by the asymmetric association measure value between two words). In *Figure 1,* is shown the directed graph obtained by using the set of words $V = \{$*isometry*, *rate of growth*, *growth rate*, *rate*$\}$ randomly extracted from WordNet where *rate of growth* and *growth rate* are synonyms, *isometry* a hyponym of the previous set and *rate* a hypernym of the same set.

The weights associated to the edges were evaluated by the Braun-Blanquet association measure (Equation 6) based on web search engine counts[2].

Figure 1 clearly shows the assumption of generality of terms as the hypernym *rate* only has incoming edges whereas the hyponym *isometry* only has outgoing edges.

As a consequence, by applying a graph-based ranking algorithm, this work aims at producing an ordered list of words from the most general (with the highest value) to the most specific (with the lowest value). For that purpose, the TextRank algorithm proposed by [12] is used both for unweighted and weighted directed graphs.

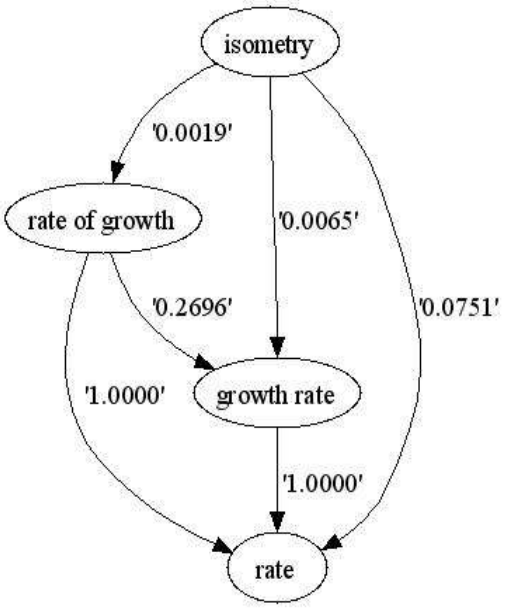

Figure 1. Directed Acyclic Graph based on the WordNet synset consisting of the words (rate of growth, growth rate) and its direct hypernym (rate) and hyponym (isometry) synsets

4.1 Unweighted Directed Graph

For a given vertex V_i let $In(V_i)$ be the set of vertices that point to it, and let $Out(V_i)$ be the set of vertices that vertex V_i points to. The score of a vertex V_i is defined in Equation 8 where d is a damping factor that can be set between 0 and 1, which has the role of integrating into the model the probability of jumping from a given vertex to another random vertex in the graph[3].

[2] We used counts returned by http://www.yahoo.com.

[3] d is usually set to 0.85.

$$S(V_i) = (1-d) + d \times \sum_{Vj \in In(V_i)} \frac{1}{|Out(V_j)|} \times S(V_j) \quad (8)$$

4.2 Weighted Directed Graph

In order to take into account, the edge weights, a new formula is introduced in Equation 9.

$$WS(V_i) = (1-d) + d \times \sum_{Vj \in In(V_i)} \frac{w_{ji}}{\sum_{Vk \in Out(V_j)} w_{jk}} \times WS(V_j) \quad (9)$$

4.3 Running TextRank

After running the algorithm in both cases, a score is associated to each vertex, which represents the "importance" of the vertex within the graph.

Notice that the final values obtained after TextRank runs to completion are not affected by the choice of the initial values randomly assigned to the vertices. Only the number of iterations needed for convergence may be different. As a consequence, after running the TextRank algorithm, in both its configurations, the output is an ordered list of words from the most general one to the most specific one. The words are ordered by their respective TextRank scores. In Table 1, are shown both the lists with the weighted and unweighted versions of the TextRank based on the directed graph shown in *Figure 1*.

Unweighted		Weighted		WordNet	
S(Vi)	Word	WS(Vi)	Word	Categ.	Word
0.50	Rate	0.81	rate	Hyper.	rate
0.27	growth rate	0.44	growth rate	Synset	growth rate
0.19	rate of growth	0.26	rate of growth	Synset	rate of growth
0.15	Isometry	0.15	isometry	Hypo.	isometry

Table 1. TextRank ordered lists

The results show that asymmetric measures combined with directed graphs and graph-based ranking algorithms such as the TextRank are likely to give a positive answer to the hypothesis about the degree of generality of terms. Moreover, an unsupervised methodology is proposed for acquiring general-specific noun relationships and respectively word ordering according to that relations.

5. EXPERIMENTS AND RESULTS

Evaluation can be a difficult task in Natural Language Processing. In fact, because human evaluation is time-consuming and generally subjective even when strict guidelines are provided, measures to automatically evaluate experiments must be proposed. In this section, are proposed two evaluation measures and the respective results are discussed.

5.1 Using constraints for evaluation

WordNet can be defined as applying a set of constraints to words. Indeed, if word *w* is the hypernym of word *x*, we may represent this relation by the following constraint $y \rangle x$, where $\rangle$ is the order operator stating that *y* is more general than *x*. For example, for each set of three synsets (the hypernym synset, the seed synset and the hyponym synset), a list of constraints can be established i.e. all words of the hypernym synset must be more general than all the words of the seed synset

and the hyponym synset, and all the words of the seed synset must be more general than all the words in the hyponym synset. So, if we take the synsets presented in *Table 1*, we can define the following set of constraints: {*rate › growth rate*, *rate › rate of growth*, *growth rate › isometry*, *rate of growth › isometry*}.

In order to evaluate the list of words ranked by the level of generality against the WordNet categorization, we just need to measure the proportion of constraints, which are respected as shown in Equation (10). This measure is called reliability of the TextRank algorithm or simply R_{TR}.

$$R_{TR} = \frac{\#\ of\ common\ constraints}{\#\ of\ constraints} \qquad (10)$$

For example, in *Table 1*, all the constraints are respected for both weighted and unweighted graphs, giving 100% R_{TR} for the ordered lists compared to WordNet categorization.

5.2 Word relations discovery by Clustering

Another way to evaluate the quality of the ordering of words is to apply hard clustering to the words weighted by their level of generality (respectively their TextRank scores). By evidencing the quality of the mapping between the hard clusters generated automatically and the corresponding WordNet *chain* of synsets, we are able to measure the quality of the ranking. In the former sentence by *chain*, is meant the path connecting WordNet synsets by the hypernymy/hyponymy relation. For example if we look at *Table 1* {rate}->{growth rate, rate of growth}->{isometry} is one such chain or path derived from WordNet. The following tasks are proposed:

(1) Perform k-means clustering[4] over the whole list of ranked words (or what is considered as the "flat" case)

(2) Instead of clustering the whole list, first take a sub-list corresponding to a part of the chain, or with other words a sub-chain, run TextRank over it and then apply k-means with a k smaller than the overall length of the original chain. After that cut the first resulting cluster (the one supposed to be the most general) and keep it. Then we add one more synset from the original chain to the remaining clusters and iteratively repeat the process until we reach the end of the given chain.

(3) Order the resulting clusters by level of generality

(4) Measure the precision, recall and *F*-measure of each cluster sorted by level of generality with its corresponding WordNet synset and compare the results from (1) and (2).

For the first and second tasks, is used the implementation of the k-means algorithm from the NLTK toolkit[5]. In particular, the k-means is bootstrapped by choosing the initial means as follows: for the first mean, is chosen the weight (the score) of the first word in the TextRank generated list of words. Then we divide the length of the list by the number of clusters that we want to create and use the resulting number as a step for choosing the next mean and so on until we choose all the means from the scores in the list.

For the third task the level of generality of each cluster is evaluated by the average level of generality of words inside the cluster (or said with other words by the mean of the scores of the words in the cluster).

[4] We are clustering the numerical scores of the words computed by using TextRank. Also other clustering algorithms can be used like Partitioning Around Medoids (PAM) or Expectation-Maximization (EM)

[5] http://nltk.sourceforge.net/

For the fourth task, the *F*-measure is considered as the reliability of the methodology and is denoted as R_T or the total reliability of the method. It is important to note that this measure includes in itself the R_{TR} and R_{KM}, where R_{KM} is the reliability of the k-means algorithm, which in our case is not directly measureable but can be computed by using the R_T and R_{TR} values, which are already known and Equation (14) shows the relation between R_T, R_{TR} and R_{KM}. Equation (14) is used to compute R_{KM} from the values of R_T and R_{TR}. The two parts of the methodology – the TextRank algorithm and k-means clustering algorithm are considered as connected in a sequential reliability scheme because the output of the TextRank is used as an input of the *k*-means clustering algorithm. The most general cluster and its corresponding WordNet synset are compared in terms of precision, recall and R_T as shown in Equation (11), (12) and (13)[6]. The same process is applied to the second most general cluster and the next synset in the WordNet chain in consideration, and so on iteratively until we reach the last cluster respectively WordNet synset in a given chain.

$$precision = \frac{|Cluster \cap Synset|}{|Cluster|} \quad (11)$$

$$recall = \frac{|Cluster \cap Synset|}{|Synset|} \quad (12)$$

$$R_T = \frac{2.recall.precision}{precision + recall} \quad (13)$$

$$R_T = R_{TR}.R_{KM} \quad (14)$$

5.3 Experiments

In order to evaluate this methodology, hypernym-hyponym chains of synsets or Taxonomic Paths from WordNet with length ranging from 2 to 10 were randomly[7] extracted. For each length were extracted 1000 distinct sample chains or overall around 15000 distinct WordNet synsets

from roughly 117,000 synsets present in the WordNet taxonomy.

For each chain of synsets, were then built the associated directed weighted and unweighted graphs using the association measures referred in Section 2[8] and the TextRank algorithm was ran to produce a general-specific ordered lists of words. After this was applied k-means to the TextRank ordered lists of words and then the bootstrapping technique with k from 2 to 5.

In *Figure 2*, are presented the results of running TextRank evaluated by the R_{TR} measure for *weighted* graphs depicting the performance of all asymmetric association measures used to construct the TextRank graphs compared to the Baseline, which is the list of ordered words from a given chain of synsets ordered by their respective Web hits frequency. As we can see from the results the Baseline is the same or better than the weighted TextRank for all asymmetric measures in consideration.

---

[6] Where $|Cluster \cap Synset|$ means the number of words common to both Synset and Cluster, and |Synset| and |Cluster| respectively measure the number of words in the Synset and the Cluster.

[7] 98% significance level is guaranteed for an error of 0.05 following the normal distribution.

[8] The probability functions are estimated by the Maximum Likelihood Estimation (MLE).

5.4 Results evaluation by Constraints (R_{TR})

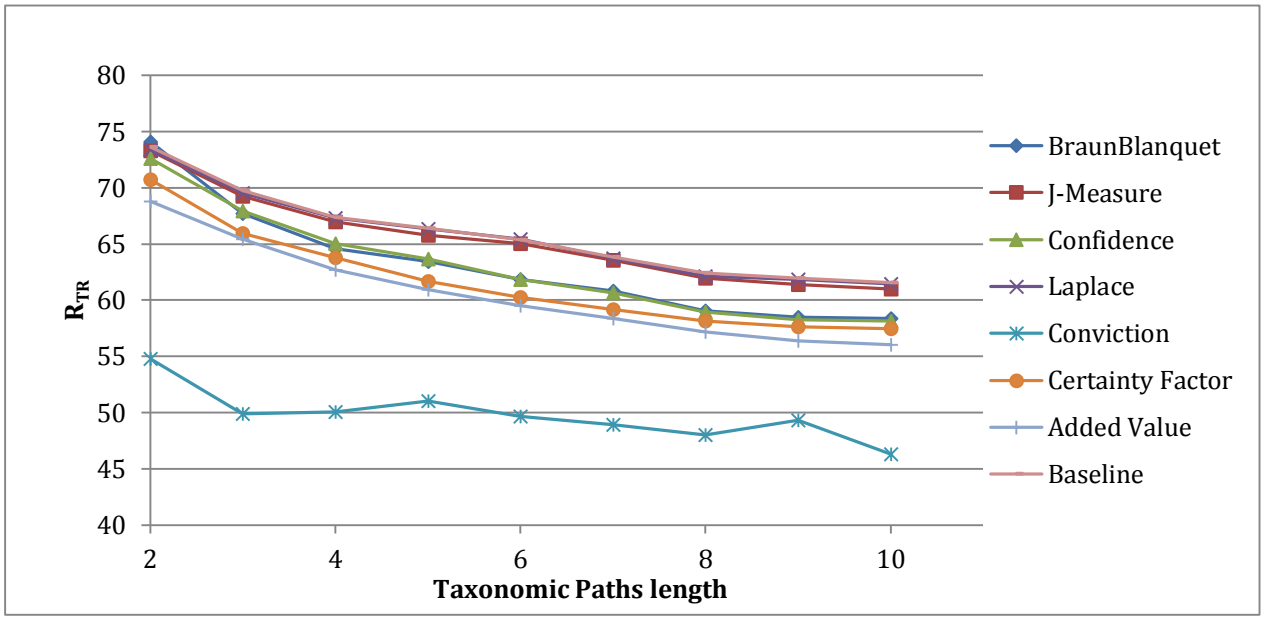

Figure 2. R_{TR} for running *weighted* TextRank with different chain lengths

It looks like in our case using *weighted* TextRank gives worse results than the *unweighted* version. Probably this is because the weights computed by the different asymmetric measures are often very small numbers and this has negative impact on the performance of the algorithm.

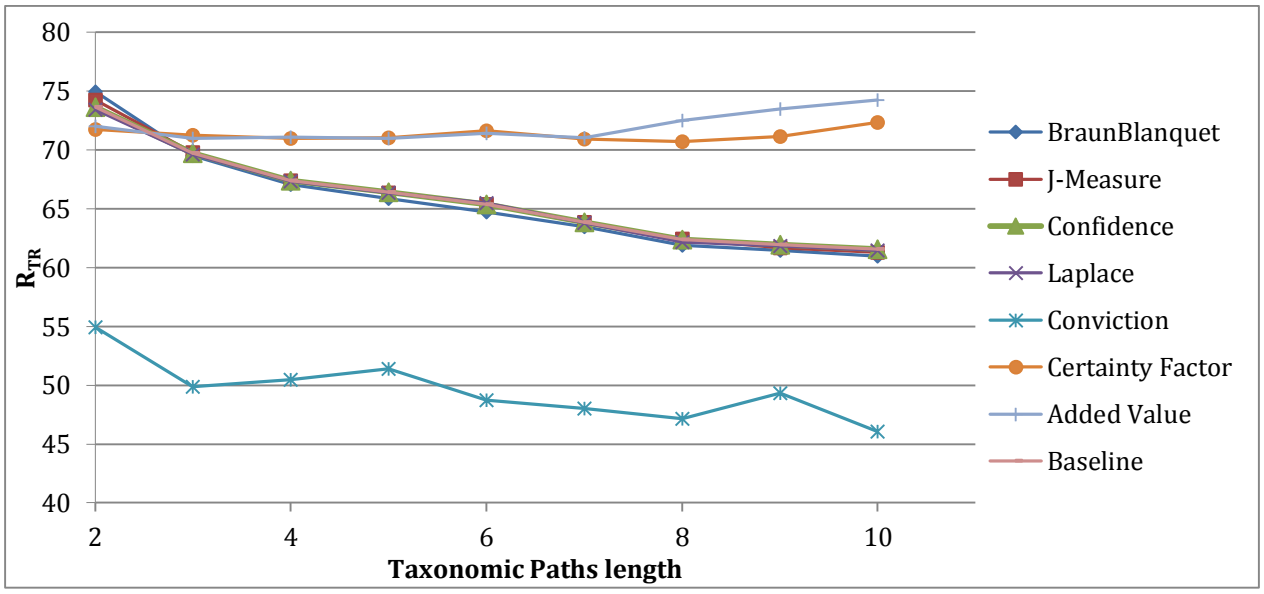

Figure 3. R_{TR} for running *unweighted* TextRank with different chain lengths

As we can see from *Figure 3* the two best performing measures are Certainty Factor and Added Value, all other measures and the Baseline have almost the same performance and the Conviction is the worst performing measure in terms of R_{TR}. It is evident that the gain of using Certainty Factor and Added Value is bigger when the chains are longer or said with other words when the WordNet Taxonomic Paths in consideration are deeper. Based on these results were selected Certainty Factor

and Added Value as the asymmetric measures to be used when running TextRank with unweighted graphs and then applying k-means as a second step in order to detect general-specific noun relationships.

5.5 Results evaluation by the clustering of TextRank scores

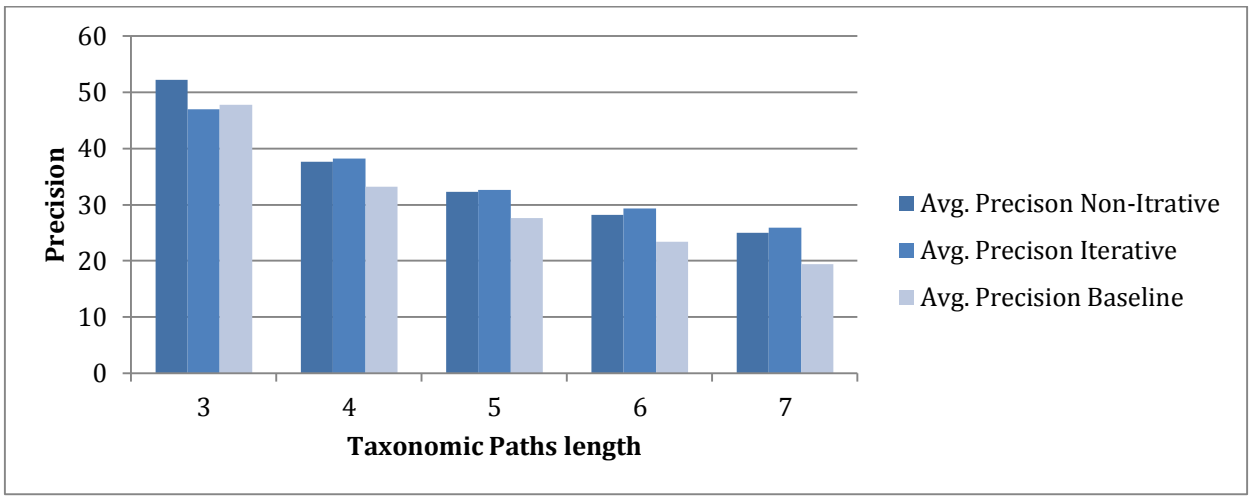

Figure 4. Certainty Factor Precision for the Non-Iterative k-means compared to the Iterative k-means and the Baseline – k-means over the word Web frequencies

In *Figure 4* is presented the average precision after applying Non-iterative or "flat" *k*-means and the results of the bootstrapping technique or the Iterative k-means with $k=2$ for sub-chain clustering compared to the Baseline. The Baseline is *k*-means applied directly to the word Web frequencies instead of the TextRank scores. For the experiment the Certainty Factor was used as an asymmetric measure to construct the graphs. As we can see from the graphic the bootstrapping was able to slightly improve the performance of the methodology in terms of precision.

6. DISCUSSION OF THE RESULTS

An important remark needs to be made at this point of the method explanation. There is a large

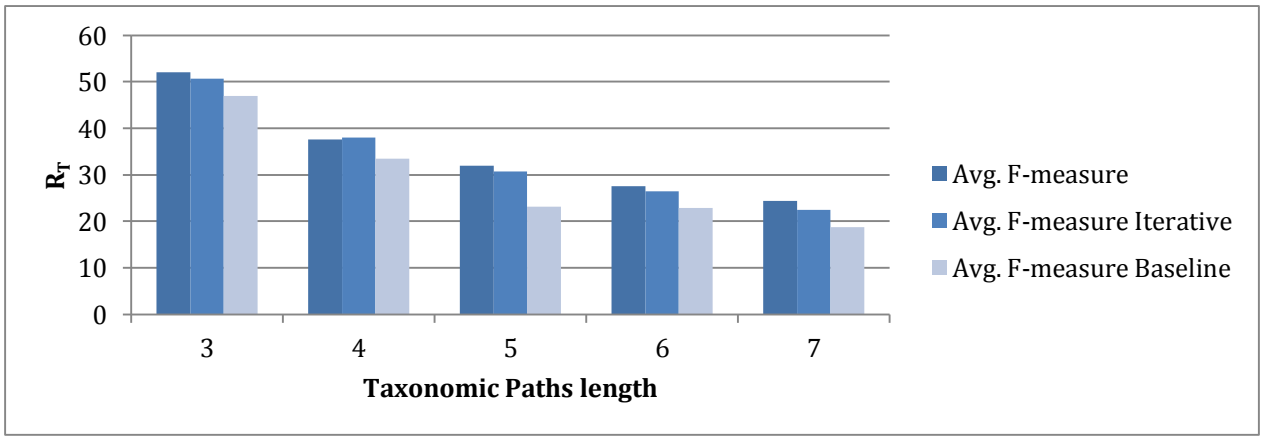

Figure 5. Average R_T for the mapping between k-means clusters and WordNet chains of synsets using Certainty Factor measure

ambiguity introduced in the methodology by just looking at web counts.

Indeed, when counting the occurrences of a word like *answer*, we count all its occurrences for all its meanings and forms. For example, based on WordNet, the word *answer* can be a verb with ten

meanings and a noun with five meanings. Moreover, words that are more frequent than others although they are not so general can skew the results.

As we are not dealing with a single domain, in which one can expect to see more often the "one sense per discourse" paradigm [13], it is clear that this methodology can perform better if restricted to a single specific domain. Now let's take a look at the average R_T measure of the mapping between clusters and WordNet synsets for both "flat" clustering and clustering with 2-level bootstrapping or Iterative *k*-means with $k=2$.

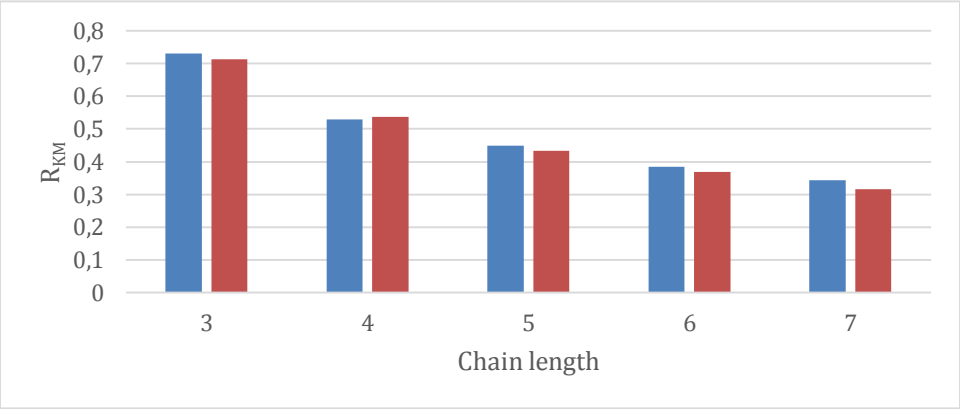

Figure 6. Average R_{KM} for the mapping between k-means clusters and WordNet chains of synsets for both Non-Iterative and Iterative experiments. Using Certainty Factor.

In *Figure 5* on the y-axis we can see the average R_T measure for the mapping of all levels of chains of a given length, which is ranging from 3 to 7 (the *x*-axis). The results in the flat and respectively the 2-level Iterative bootstrapping case (see section 5.2) are shown. As we can see from the graphic the bootstrapping technique did not improve the overall results of the mapping when R_T measure is taken into consideration. Just the precision of the algorithm was improved.

In *Figure 6* are shown the average calculated values for R_{KM}. As we can see there are fairly good results for short chains and when the chains become longer the R_{KM} deteriorates. In the case of Iterative *k*-means the R_{KM} is slightly lower (the red bars on the chart).

In addition there has been a great discussion in the corpora list[9] whether one should use web counts instead of corpus counts to estimate word frequencies. In this study, we clearly see that web counts show evident problems, like the ones mentioned by [10]. However, they cannot be discarded so easily. In particular, this work can be applied at web counts in web directories that would act as specific domains and would reduce the space of ambiguity. Of course, experiments with well-known corpora will also have to be made to understand better this phenomenon.

7. CONCLUSIONS AND FUTURE WORK

In this paper, is proposed a new methodology based on directed weighted/unweighted graphs, the TextRank algorithm and the k-means clustering to automatically bootstrap word order from web corpora frequency counts. As far as the knowledge of the author, such an unsupervised experiment has never been attempted so far. In order to evaluate the results, have been proposed different evaluation metrics showing promising results for word ordering.

As future work, the author intends to address the problem with graph construction and the low performance of the clustering algorithm. As we have seen from the results in the paper the topology

[9] Finalized by [7].

of the graph looks more important then it's weights if we use the proposed asymmetric measures. In the moment the graphs are created in such way that we always connect the nodes in the graph, even when the weight (the value of the asymmetric association measure in use) is very small. The author intends to try pruning and similar techniques in order to reduce the less important edges in the graph, which will eventually lead to an improvement in the performance of the proposed methodology. In addition, other clustering algorithms have to be tested like Expectation-Maximization (*EM*) for example. K-means assumes that the clustered values are uniformly distributed while *EM* assumes normal distribution and could give better results than *k*-means.

Another way of development is to experiment with the methodology on lists of words derived by paraphrase alignment and also on automatically made clusters of semantically similar words.

Other ideas for application of the word ordering method proposed here are to apply it in the evaluation of the Wikipedia taxonomy, which is not supervised. Creation of user mini-ontologies in Information Retrieval, Synonymy detection by generality level and automatic unsupervised taxonomy construction.

8. REFERENCES

[1] Bollegala, D., Matsuo, Y. and Ishizuka, M. 2007. Measuring Semantic Similarity between Words Using WebSearch Engines. In Proceedings of International World Wide Web Conference (WWW 2007).

[2] Caraballo S.A. 1999. Automatic Construction of a Hypernym-labeled Noun Hierarchy from Text. In Proceedings of the Conference of the Association for Computational Linguistics (ACL 1999).

[3] Dias G., Mukelov R. and Cleuziou G. 2008. Unsupervised Learning of General-Specific Noun Relations from the Web. 21th International FLAIRS Conference (FLAIRS 2008). AAAI Press. Coconut Grove, Florida, USA. May 15-18. pp. 147-153.

[4] Dias G., Santos C., and Cleuziou G. 2006. Automatic Knowledge Representation using a Graph-based Algorithm for Language-Independent Lexical Chaining. In Proceedings of the Workshop on Information Extraction Beyond the Document associated to the Joint Conference of the International Committee of Computational Linguistics and the Association for Computational Linguistics (COLING/ACL), pp. 36-47.

[5] Garipova Julia, Anton Georgiev, T. Papanchev, N. Nikolov and D. Zlatev. Operational Reliability Assessment of Systems Containing Electronic Elements. 2nd International Scientific Conference "Intelligent information technologies for industry", 14-16.9.2017, Varna, Bulgaria. © Springer International Publishing AG 2018; Proceedings of the Conference IITI'17, pp.340-348 Advances in Intelligent Systems and Computing 680, DOI: 10.1007/978-3-319-68324-9_37.

[6] Georgiev Anton, Papanchev T., Nikolov N. Reliability assessment of power semiconductor devices. 19th International Symposium on Electrical Apparatus and Technologies, SIELA 2016.

[7] Georgiev Anton, Nikolov N., Papanchev T. Maintenance process efficiency when conduct reliability-centered maintenance of complex electronic systems. 19th International Symposium on Electrical Apparatus and Technologies, SIELA 2016.

[8] Grefenstette G. 1994. Explorations in Automatic Thesaurus Discovery. Kluwer Academic Publishers, USA.

[9] Hearst M. H. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. In Proceedings of the Fourteenth International Conference on Computational Linguistics (COLING 1992), pages 539-545.

[10] Kilgarriff A. 2007. Googleology is Bad Science. Computational Linguistics 33 (1), pp 147-151.

[11] Michelbacher L., Evert S. and Schütze H. Asymmetric Association Measures. In Proceedings of the Recent Advances in Natural Language Processing (RANLP 2007).

[12] Mihalcea R. and Tarau P. 2004. TextRank: Bringing Order into Texts. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004), pp 404-411.

[13] Moralyiski R., and Dias G. (2007). One Sense per Discourse for Synonym Detection. International Conference On Recent Advances in Natural Language Processing (RANLP 2007). Borovets, Bulgaria, September 27-29. ISBN: 978-954-91743-7-3. pp. 383-387.

[14] Pecina P. and Schlesinger P. 2006. Combining Association Measures for Collocation Extraction. In Proceedings of the International Committee of Computational Linguistics and the Association for Computational Linguistics (COLING/ACL 2006).

[15] Riloff E. 1993. Automatically Constructing a Dictionary for Information Extraction Tasks. In Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI 1993), pages 811-816.

[16] Sang E.J.K. and Hofmann K. 2007. Automatic Extraction of Dutch Hypernym-Hyponym Pairs. In Proceedings of Computational Linguistics in the Netherlands Conference (CLIN 2007).

[17] Snow R., Jurafsky D. and Ng A. Y. 2006. Learning Syntactic Patterns for Automatic Hypernym Discovery. In Proceedings of the International Committee of Computational Linguistics and the Association for Computational Linguistics (COLING/ACL 2006).

[18] Snow R., Jurafsky D. and Ng A. Y. 2005. Semantic Taxonomy Induction from Heterogenous Evidence. In Proceedings of the Neural Information Processing Systems Conference (NIPS 2005).

[19] Stevenson M., and Greenwood M. 2006. Comparing Information Extraction Pattern Models. In Proceedings of the Workshop on Information Extraction Beyond the Document associated to the Joint Conference of the International Committee of Computational Linguistics and the Association for Computational Linguistics (COLING/ACL 2006), pages. 29-35.

[20] Tan P.-N., Kumar V. and Srivastava, J. 2004. Selecting the Right Objective Measure for Association Analysis. Information Systems, 29 (4)., pages 293-313.

[21] Sanderson M. and Croft B. 1999. Deriving concept hierarchies from text. In proceedings of the Annual ACM Conference on Research and Development in Information Retrieval, pages 206-213.