Extraction of Semantic Relations between Concepts with KNN Algorithms on Wikipedia

Alexander Panchenko[1,2], Sergey Adeykin[2], Alexey Romanov[2], and Pavel Romanov[2]

[1] Université catholique de Louvain, Centre for Natural Language Processing, Belgium
[2] Bauman Moscow State Technical University, Information Systems dept. (IU5), Russia
{panchenko.alexander,adeykin90,jgc128ra,romanov4400}@gmail.com

Abstract. This paper presents methods for extraction of semantic relations between words. The methods rely on the k-nearest neighbor algorithms and two semantic similarity measures to extract relations from the abstracts of Wikipedia articles. We analyze the proposed methods and evaluate their performance. Precision of the extraction with the best method achieves 83%. We also present an open source system which effectively implements the described algorithms.

Keywords: semantic relations, information extraction, Wikipedia, KNN, MKNN, semantic similarity measure, computational lexical semantics

1 Introduction

There exist many types of semantic relations between words (concepts) – synonymy, metonymy, antonymy, association, etc. In the context of this work, semantic relations are synonyms, hypernyms, and co-hypernyms (words with a common hypernym). These relations are successfully used in various NLP applications, such as word sense disambiguation [1], query expansion [2], document categorization [3] or question answering [4]. Semantic relations are fixed manually in various linguistic resources, such as thesauri, ontologies, and synonym dictionaries. However, existing resources are often not available for a given NLP application, domain, or language. Furthermore, manual construction of the required semantic resources is an extremely expensive and time-consuming process. This motivates the development of new relation extraction methods.

A popular approach to relation extraction is based on the lexico-syntactic patterns [5]. The main drawbacks of this approach are complexity of pattern construction and their language dependency. Methods, based on the distributional analysis [6, 7], do not require any manual labor, but are less precise [8]. Recently, the measures of semantic similarity based on Wikipedia[1] have been proposed [9, 10 ,11]. Wikipedia is attractive for text mining as it contains articles about all the main domains written in all the main languages. Furthermore, it is constantly updated by users. Wikipedia-

[1] Wikipedia, the free encyclopedia that anyone can edit: http://www.wikipedia.org/

based measures show excellent results on the task of correlation with human judgments. Until now, these measures were not used to extract semantic relations.

The approach described in this work fills this gap, and focuses on the application of Wikipedia-based similarity measures to semantic relation extraction. The goal of the method proposed in this article is to discover a set of relations R between a set of input concepts C (e. g. terms of a given domain). In this work, we deal with linguistic concepts, such as words or multiword expressions (not to be confused with a similar notion from the Formal Concept Analysis [23]). The proposed method does not return the type of the relationship between words, i. e. $R \subseteq C \times C$. The method is computationally efficient, sufficiently precise for the practical NLP applications, and can be applied for the languages available in Wikipedia. The main contributions of this paper are the following:

1. The new semantic relation extraction methods, which rely on the texts of Wikipedia articles, k-nearest neighbors algorithms (KNN and MKNN), and two similarity measures (Cosine and Gloss Overlap).
2. An Open Source system Serelex (LGPLv3), which efficiently implements the proposed methods.

In Section 2, we introduce our approach to semantic relation extraction. First, we describe the data and how they are preprocessed in Section 2.1. Next, we discuss the algorithms of semantic relation extraction (Section 2.2) and the used measures of semantic similarity (Section 2.3). Finally, we present key details of the extraction system Serelex (Section 2.4). In Section 3, the experimental results are presented and discussed. Section 4 deals with the related work and Section 5 wraps up with a conclusion and a description of the future research.

2 Semantic Relation Extraction Methods

2.1 Data and Preprocessing

Input data of the method is a set of definitions D for each input concept $c \in C$. We use the data available from the DBPedia.org to build a set of definitions of English terms (multi-word expressions are not included)[2]. For each input concept a pair (c, d) is built, where concept c is an exact title of a Wikipedia article, and definition d is a text of the first paragraph of this article. The experiments described in this work were conducted on a subset of articles with titles containing no numbers and special symbols. We collected 327.167 Wikipedia articles according to this principle. For the goals of our experiments, we prepared two datasets containing 775 words (824Kb) and 327.167 words (237Mb) respectively[3].

Articles were preprocessed as follows. First, we removed all markup tags and special characters. Second, we performed lemmatization and part-of-speech tagging with the TreeTagger [12]. As a result, each word was represented as a triple "to-

[2] We used the file http://downloads.dbpedia.org/3.7/en/long_abstracts_en.nt.bz2
[3] Data are available at: http://cental.fltr.ucl.ac.be/team/~panchenko/def/

ken#POS#lemma", for instance "proved#VVN#prove". An example of a definition in this format is provided below:

axiom; in#IN#in traditional#JJ#traditional logic#NN#logic ,#,#, an#DT#an axiom#NN#axiom or#CC#or postulate#NN#postulate is#VBZ#be a#DT#a proposition#NN#proposition that#WDT#that is#VBZ#be not#RB#not proved#VVN#prove or#CC#or demonstrated#VVN#demonstrate but#CC#but considered#VVN#consider to#TO#to be#VB#be either#RB#either self-evident#JJ#self-evident ,#,#, or#CC#or subject#JJ#subject to#TO#to necessary#JJ#necessary decision#NN#decision .#SENT#.

Senlar [19] and other researchers [7] mention that the methods based on the syntactic analysis achieve higher results than the methods based only on the part-of-speech tagging. However, in our method we intentionally do not use the syntactic analysis for two reasons. Firstly, the computational complexity of the parsing algorithms is very high. Secondly, such a complex linguistic analysis makes the method less robust. Prior researches suggest that quality of parsing in different languages is very different [22]. Moreover, the standard parsers make a lot of errors in the sentences which contain named entities and technical terms, the lexical units which are the most valuable in the context of information extraction.

2.2 Algorithms of Semantic Relation Extraction

Algorithms of semantic relation extraction discussed in this article, are based on the component analysis [13, 14], which states that semantically similar words have similar definitions. The proposed methods use one of two similarity measures: Gloss Overlap of definitions [15] or Cosine between vectors of definitions [16]. The method takes as an input a set of concepts C and outputs a set of relations R between them. Assume that the algorithm is processing the 5 following concepts: $C = \{alligator, animal, building, house, telephone\}$. Its goal would be to recognize the set of semantic relations $R = \{\langle alligator, animal\rangle, \langle building, house\rangle\}$ out of 10 possible pairs of concepts.

The first algorithm calculates semantic relations with the KNN algorithm, the second relies on the MKNN (Mutual KNN) algorithm. The only meta-parameter of the algorithms is the number of nearest neighbors k. The algorithms are presented in Fig.1.

The algorithms first calculate pairwise similarities between all the input terms (lines 1-12). The array of the nearest neighbors `Rmatrix` contains nearest neighbors of each term. We keep the number of elements in each row of `Rmatrix` equals k, which let us minimize the memory footprint. The last stage for the KNN is simply to print the k nearest neighbor concepts for each concept. In contrast to KNN, MKNN establishes a relation only if a pair of words (c_i, c_j) are mutual neighbors (lines 13-21). Therefore, MKNN filters out those relations extracted by KNN which are not mutually related.

Complexity of the developed algorithms is a function of the number of input concepts $|C|$. Time complexity is $O(|C^2|)$ and space complexity is $O(k|C|)$, where k is the number of nearest neighbors.

```
   R = ComponentAnalysis(C, D, k, isMutualKNN)
   Input: C – concepts, D – definitions of concepts, k – number of nearest
   neighbors, isMutualKnn – if true then MKNN, else KNN
   Output: R – set of semantic relations <c_i,c_j> in C X C
1. // Calculation of pairwise similarities between words all concepts C
2. Rmatrix = void
3. for i=0; i<count(C); i++ {
4.       for j=i; j<count(C); j++ {
5.             // Calculation of semantic similarity of two concepts
6.             s_ij = similarity(D(i), D(j))
7.             // Saving most similar concepts
8.             if( count(Rmatrix(C(i))) < k || s_ij > min(Rmatrix(C(i)) ){
9.                   Rmatrix(C(i)).addOrReplaceMin(C(j))
10.            }
11.      }
12. }
13. // Calculation of semantic relations
14. R = void
15. foreach c_i in Rmatrix {
16.      foreach c_j in Rmatrix(c_i) {
17.            if(!isMutualKNN || Rmatrix(c_j) contains c_i){
18.                  R.add(<c_i, c_j>)
19.            }
20.      }
21. }
22. return R
```

Fig. 1. Semantic relation extraction algorithms KNN and MKNN.

2.3 Measures of Semantic Similarity

Function `similarity` (line 6) in the algorithms KNN and MKNN calculates a pairwise semantic similarity of two concepts $c_i, c_j \in C$, from their definitions $d_i, d_j \in D$. The larger the value of semantic similarity, the closer the "sense" of the concepts. Two similarity functions are considered here. The first is the gloss overlap of the definitions d_i, d_j of the concepts c_i, c_j:

$$similarity(c_i, c_j) = \frac{2|d_i \cap d_j|}{|d_i| + |d_j|}.$$

Here the numerator is the number of the common words in the definitions; $|d_j|$ is the number of words in the definition d_j. The second measure is the cosine between vectors $\boldsymbol{f}_i, \boldsymbol{f}_j$ of definitions d_i, d_j representing concepts c_i, c_j:

$$similarity(c_i, c_j) = \frac{\boldsymbol{f}_i \cdot \boldsymbol{f}_j}{\|\boldsymbol{f}_i\| \, \|\boldsymbol{f}_j\|} = \frac{\sum_{k=1,N} f_{ik} f_{jk}}{\sqrt{\sum_{k=1,N} f_{ik}^2} \sqrt{\sum_{k=1,N} f_{jk}^2}}.$$

Here f_{ik} is the frequency of the lemma c_k in the definition d_i. The both similarity measures use the lemmas (e. g., animals#NNS#animal), and do not use stopwords found in the definitions. For the both of similarity measures only matches of the nouns (NN, NNS, NP), verbs (VV, VVN, VVP), and adjectives (JJ) are considered.

2.4 Relation Extraction System Serelex

The system is a console application implemented in C++ and available for Windows and Linux platforms (32/64 bits). It consists of the definition class, global parameters class, component analysis class, and several additional classes and functions (see Fig.2). The main functions of the program are:

- loading files of stopwords and input concepts C;
- loading the file with definitions D taking into account the stopwords;
- calculation of the pairwise semantic similarities between the input concepts C;
- building the list of the semantic relations R.

In order to achieve high performance of the system, we map each word to a numerical identifier. This procedure significantly reduces running time of the program. The system extensively uses the Standard Template Library (STL) and the Boost library[4]. The source code and the binaries of the Serelex are available under the conditions of the LGPLv3 license at https://github.com/AlexanderPanchenko/Serelex.

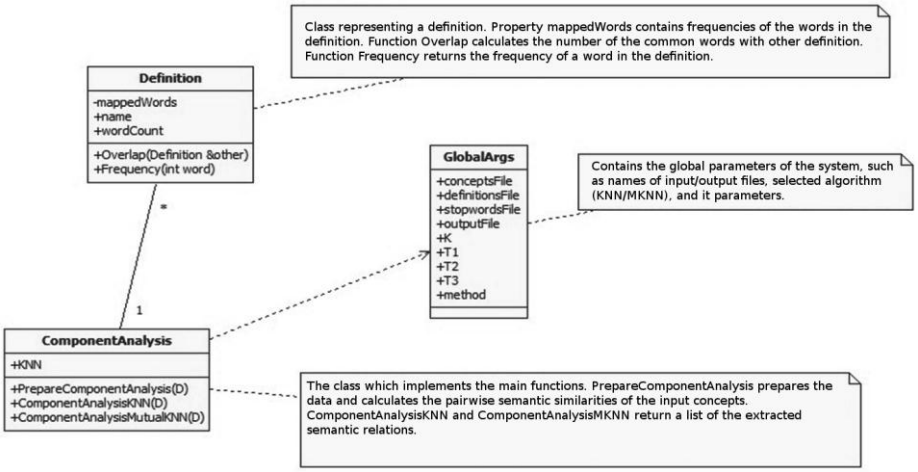

Fig. 2. The main classes of the semantic relation extraction system Serelex.

3 Results

We investigated the algorithms KNN and MKNN with the two measures described above and with various numbers of nearest neighbors k (see Fig. 3). As one may expect, the number of extracted relations linearly depends on the number of nearest neighbors k both for KNN and MKNN. The number of extracted relations depends little on the similarity measure type. The key difference between the two measures is that Cosine takes into account frequencies, while Gloss Overlap does not. The little

[4] Boost C++ libraries: http://www.boost.org/

difference in the results is likely to be due to the fact that the definitions are short. Thus, frequency information does not contribute a lot to the result.

The algorithm KNN extracts more relations than the MKNN for the same value of k. It happens because the MKNN filters out pairs of concepts which are not mutual nearest neighbors. According to our experiments, MKNN filters around 50-70% of the relations extracted by KNN, depending on the number of the concepts $|C|$ and the value of k.

We estimated the precision of the extraction between a set of 775 concepts for both algorithms with $k = 2$. In order to measure the precision, we manually labeled the files with the extracted relations. The precision was calculated as the number of correctly extracted relations to the number of extracted relations. The results are presented in Table 1. The examples of extracted relations between a set of 775 concepts with algorithm MKNN ($k = 2$) and the gloss overlap similarity measure are presented below[5]:

$R = \{\langle acacia, pine\rangle, \langle aircraft, rocket\rangle, \langle alcohol, carbohydrate\rangle,$
$\langle alligator, coconut\rangle, \langle altar, sacristy\rangle, \ldots \langle object, library\rangle,$
$\langle object, pattern\rangle, \langle office, crew\rangle, \langle onion, garlic\rangle, \langle saxophone, violin\rangle,$
$\langle saxophone, clarinet\rangle \langle tongue, mouth\rangle, \ldots, \langle watercraft, boat\rangle,$
$\langle watermelon, berry\rangle, \langle weapon, warship\rangle, \langle wolf, coyote\rangle, \langle wood, paper\rangle\}.$

Due to the huge number of extracted relations (see Fig. 3), it is hard to calculate manually the extraction precision for all values of k. We expect the precision to decrease for values of $k > 2$. According to our preliminary experiments, we recommend to use the number of the nearest neighbors $k \in [1; 10]$. In future, we plan to use the WordNet [17] and the standard evaluation datasets, such as BLESS [18], to estimate the precision of the method more robustly.

The performance of the developed system is rather high. For instance, the file with 775 definitions is processed for less than 3 seconds on a Linux server equipped with Intel(R) Xeon(R) CPU E5606@2.13GHz (the program does not use multithreading). The algorithm KNN used with the measure Gloss Overlap processes the file with the 327.167 definitions within 3 days 3 hours and 47 minutes.

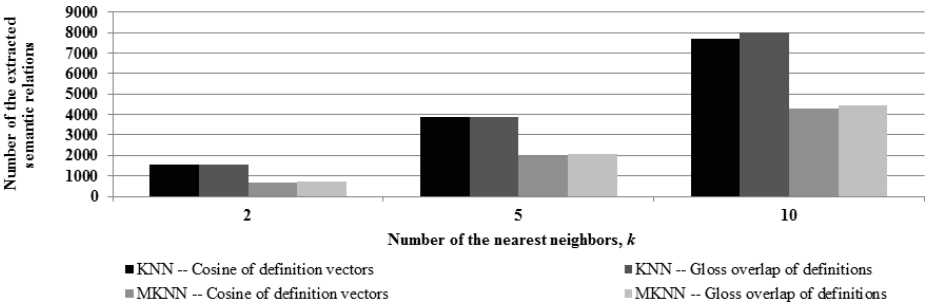

Fig. 3. Dependence of the number of extracted relations on the number of nearest neighbors k.

[5] The full list of the extracted relations with this configuration is available at http://cental.fltr.ucl.ac.be/team/~panchenko/def/results-775/overlap_mknn_2.csv

Table 1. Precision of relation extraction for 775 concepts with the KNN and MKNN ($k = 2$).

Algorithm	Similarity Measure	Extracted Relations	Correct Relations	Precision
KNN	Cosine	1548	1167	0.754
	Gloss overlap	1546	1176	0.761
MKNN	Cosine	652	499	0.763
	Gloss overlap	724	603	**0.833**

4 Related Work

Senlar [19] presents a recent overview of semantic relation extraction methods based on text corpora and electronic dictionaries. The automatic thesaurus construction system SEXTANT extracts relations between words with precision around 75%. Measures of semantic similarity based on the Web achieve precision in TOEFL synonymy test up to 74%.

WikiRelate! is the most similar system to ours. It was proposed by Strube and Ponzetto in 2006 [9]. The main differences of our method and the system from this development are the following:

- Serelex extracts semantic relations, while WikiRelate! only implements a semantic similarity measure;
- The source code of WikiRelate! is not available, while the binary version is available only for research purposes. The source code of Serelex is available under the conditions of the industry-friendly LGPLv3 license;
- Serelex can calculate similarity not only between texts of Wikipedia, but also between any other definitions represented in the corresponding format;
- Serelex implements two measures of similarity (Cosine and Gloss Overlap), while WikiRelate! implements only the Gloss Overlap measure. Furthermore, the designers of the system implement the Gloss Overlap in a different way: they count each match of n-gram as n^2 matches;
- Serelex does not use the category lattice of Wikipedia.

WikiRelate! achieves a correlation with human judgments of 0.59. However, since WikiRelate! does not extract relations, we cannot directly compare its performance with our results.

In [10] and [11], the authors proposed alternative measures of semantic similarity based on texts of Wikipedia. However, those approaches are less similar to Serelex than the WikiRelate!. In particular, these measures represent the concepts in a vector space of all Wikipedia articles, while Serelex uses the traditional bag-of-words space. Nakayama et al. [20] suggested yet another relation extraction method based on Wikipedia, which is significantly different from our method. The authors use the hyperlinks structure of Wikipedia articles to infer associations between words. Finally, Milne et al. [21] suggested to extract synonyms, hyponyms, and associations from Wikipedia category lattice, and other structure and navigational elements of Wikipedia.

5 Conclusion

We proposed and analyzed the method for semantic relation extraction from texts of Wikipedia with algorithms KNN and MKNN and two semantic similarity measures. The preliminary experiments showed that the best results (precision of 83%) are obtained with the method based on MKNN algorithm and Gloss Overlap similarity measure. We also presented an open source system, which efficiently implements the proposed method.

The developed method seems to be enough precise and computationally efficient for the practical NLP applications. Currently, we are working on integration of the extracted semantic relations in a text categorization system [24]. Another application of the extracted relations, which we would like to investigate, is query expansion.

The method is able to calculate relations between a huge number of concepts, each of which is represented by a title of a Wikipedia article. Thus, it could be used to extract relations between 3.8 million of concepts in English Wikipedia and 17 million of concepts in other 282 languages of Wikipedia. The only language-dependent resources used in the method are stoplist, part-of-speech tagger, and lemmatizer. These resources are available for the most of the European languages for free. Finally, Serelex can extract relations from other sources of definitions, such as traditional dictionaries or Wiktionary, provided these data are provided in the proper format.

The main directions of our future research are: (1) using the developed method to extract relations in Russian, French, and German languages; (2) improving the precision of the extraction by clustering of the obtained semantic relations graph.

Acknowledgments

Yuri Philippovich, Adrien Dessy, Olga Morozova, Nicolas Danhier, and three anonymous reviewers provided comments and remarks, which considerably improved quality of the paper. Alexander Panchenko thanks Wallonie-Bruxelles International (WBI) foundation for the support.

References

1. Patwardhan, S., Pedersen, T.: Using WordNet-based context vectors to estimate the semantic relatedness of concepts. In: Proceedings of the EACL 2006 Workshop Making Sense of Sense, pp. 1-12 (2006)
2. Hsu, M.H., Tsai, M.F., Chen, H.H.: Query expansion with conceptnet and wordnet: An intrinsic comparison. In: Information Retrieval Technology, pp. 1–13, Springer (2006)
3. Tikk, D., Yang, J.D., Bang S.L.: Hierarchical text categorization using fuzzy relational thesaurus. In: KYBERNETIKA-PRAHA, vol. 39(5), pp.583–600 (2003).
4. Sun, R., Jiang, J., Fan, Y., Hang, T., Tatseng, K., Yen Kan, C.M.: Using syntactic and semantic relation analysis in question answering. In: Proceedings of the TREC (2005)
5. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora, In: Proceedings of the 14th conference on Computational linguistics COLING '92, pp.539-545 (1992)

6. Lin D. Automatic retrieval and clustering of similar words. In: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics, pp.768-774 (1998)
7. Heylen, K., Peirsman ,Y., Geeraerts, D., Speelman, D.: Modelling word similarity: an evaluation of automatic synonymy extraction algorithms. In: Proceedings of the Sixth International Language Resources and Evaluation (LREC'08), pp. 3243-3249 (2008)
8. Curran, J.R., and Moens, M.: Improvements in automatic thesaurus extraction. In: Proceedings of the ACL-02 Workshop on Unsupervised Lexical Acquisition. pp. 59-66 (2002)
9. Strube, M., and Ponzetto, S.P.: WikiRelate! Computing semantic relatedness using Wikipedia. In: Proceedings of the National Conference on Artificial Intelligence, pp. 1419-1429, (2006)
10. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: International Joint Conference on Artificial Intelligence, pp. 12-20 (2007)
11. Zesch, T., Müller, C., Gurevych, I.: Extracting lexical semantic knowledge from wikipedia and wiktionary. In: Proceedings of the LREC, pp. 1646–1652, 2008.
12. Schmid, H..: Probabilistic Part-of-Speech Tagging Using Decision Trees. pp. 44–49, (1994)
13. Philippovich, Y.N., Prokhorov, A.V.: Semantika informatsionnih tehnologii: opiti slovarno-tezaurusnogo opisaniya (In Russian). Series «Computational Linguistics». M.:MGUP, http://it-claim.ru/Library/Books/CL/CLbook.htm (2002)
14. Kobozeva, I.M.: Komponentnii analiz leksicheskogo znacheniya (In Russian). In: Linguistical Semantics: 4rd ed.. M.: Edition «LIBRICOM», pp. 109-122 (2009)
15. Banerjee, S., Pedersen, T.: Extended Gloss Overlaps as a Measure of Semantic Relatedness,In: Proceedings of the International Joint Conference on Artificial Intelligence (2003)
16. Jurafsky, D., Manning, H. M.: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Second Edition. pp.697-701, 2009.
17. Fellbaum, C.: WordNet. Theory and Applications of Ontology: Computer Applications, pp. 231-243, Springer (2010)
18. Baroni, M., and Lenci, A.: How we BLESSed distributional semantic evaluation. In Proceedings of GEMS Workshop of EMNLP 2011 (2011)
19. Senellart, P., Blondel, V. D.: Automatic Discovery of SimilarWords. In: Survey of Text Mining II. vol. 1, pp. 25-44, Springer London (2008)
20. Nakayama, K., Hara, T., and Nishio S.: Wikipedia Mining for an Association Web Thesaurus Construction. In: Web Information Systems Engineering – WISE, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 322-334 (2007)
21. Milne, D., Medelyan, O., and Witten, I.H.: Mining Domain-Specific Thesauri from Wikipedia: A Case Study. In: Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, pp. 442-448, IEEE Computer Society (2006)
22. Candito, M., Nivre, J., Denis, P. Anguiano, E. H.: Benchmarking of statistical dependency parsers for French. In: Proceedings of the 23rd International Conference on Computational Linguistics COLING '10. Stroudsburg, PA, USA, pp.108-116 (2010)
23. Ganter, B., and Wille, R., and Franzke, C.: Formal concept analysis: mathematical foundations. Springer-Verlag New York (1997)
24. Panchenko, A., Beaufort, R., Fairon, C.: Detection of Child Sexual Abuse Media on P2P Networks: Normalization and Classification of Associated Filenames. In: Proceedings of Workshop on Language Resources for Public Security Applications. The 8th International Conference on Language Resources and Evaluation (LREC), Istambul, Turkey (2012