Modeling of Lexical Relations Between Topics Retrieved from DBLP Journals

Lukáš Hlaváček and Michal Výmola

Department of Computer Science, FEI, VSB - Technical University of Ostrava,
17. listopadu 15, 708 33, Ostrava-Poruba, Czech Republic
{lukas.hlavacek, michal.vymola}@vsb.cz

Abstract. In this paper we present a method for getting topics from aims and scopes in DBLP journals and following construction of their hierarchical order. We focused on semantic relations between topics. Our method is fully automatic but manual cleaning of topic database would lead to much better accuracy. Another purpose of our work is to provide similar system to ACM classification system. We want to provide better, newer and fully automatic system contrary to ACM.

Keywords: DBPL, ACM, Lexical acquisition, Text mining

1 Introduction

This paper describes a method for semantically retrieved topics hierarchy into a graph where the nodes are topics and the edges (also called links) represent relationships between them. We present how to retrieve data from plain text, then algorithm itself an in the last part testing and result analysis. There were not well presented yet methods for automatic topic retrieval from plain text. Of course, our technique is applicable to many other issues requiring word categorization with semantic influence. Existing methods are primarily focused on semantic of ordinary words. In next chapter we mention some of them. But we want to focus in this paper especially on semantic of topics because topics have a special semantic. This paper describes a method for semantically retrieved topics hierarchy into a graph where the nodes are topics and the edges (also called links) represent relationships between them. Levels in our graph represent level in hierarchy.

Our motivation was Widdow's method described in [1], but we apply his technique on topics retrieved from DBLP. For testing we could not use WordNet database as Widdows did because this database contains only information about simple words (synonyms). Topics mostly consists of more words (2-3). That is why we choose ACM classification (latest 1998).

ACM is widely recognized as the premier membership organization for computing professionals, delivering resources that advance computing as a science and a profession, enable professional development and promote policies and research that benefit society.

V. Snášel, J. Pokorný, K. Richta (Eds.): Dateso 2011, pp. 153–160, ISBN 978-80-248-2391-1.

ACM hosts the computing industry's leading Digital Library and Guide to Computing Literature, and serves its global members and the computing profession with journals and magazines, conferences, workshops, electronic forums, and Online Books and Courses.

ACM's first classification system for the computing field was published in 1964. Then, in 1982, the ACM published an entirely new system. New versions based on the 1982 system followed, in 1983, 1987, 1991, and 1998 [8]. ACM classification is already more than 10 years old and does not contain many new topics, for instance *social network* or *wireless networks*.

The DBLP data source, which is representative of conventional database applications, is maintained by a single source. It is one of the best formatted and organized bibliography datasets. DBLP covers approximately 400,000 researchers who have publications in major Computer Science publication venues. Bibliographic datasets have been used for social network analysis, such as studying the structure and the spread of influence in scientific communities.

2 Previous work

In this paper we continue in D. Widdows' work. He presented method for assembling semantic knowledge from a part-of-speech tagged corpus using graph algorithms. His graph model is built by linking pairs of words which participate in particular syntactic relationships.

In this part we present some of most important previous methods. Most work on automatic lexical acquisition has been based at some point on the notion of semantic similarity. Roark and Charniak described a *generic algorithm* for extracting such lists of similar words using the notion of semantic similarity in [7].

Roark and Charniak reported accuracies of 17% and 35%. The early results have been improved upon by Riloff and Jones in [6], where a *mutual bootstrapping* approach is used to extract words in particular semantic categories and expression patterns for recognizing relationships between these words for the purposes of information extraction. The accuracy achieved in this experiment was about 78%.

Another way to obtain word-senses directly from corpora is to use clustering algorithms on feature-vectors. General problem for such clustering techniques lies in the question of how many clusters one should have, i.e. how many senses are appropriate for a particular word in a given domain. Lin's approach to this problem in [10] is to build a *similarity tree* (using what is in effect a hierarchical clustering method) of words related to a target word. Another approach described T. P. Martin and M. Azmi-Murad in [8].

Widdows described a new incremental algorithm for extracting categories of similar words. He defined following method for constructing a hierarchy of words, affecting how they depend on each other.

Definition 1. *Let A be a set of nodes and let $N(A)$ be the neighbours of A. These neigbour nodes are linked to any $a \in A$. (So $N(A) = \bigcup_{a \in A} N(a)$.) The* *best new node is taken to be the node $b \in N(A) \setminus A$ with the highest proportion of links to N (A). More precisely, for each $u \in N(A) \setminus A$, let the affinity between u and A be given by the ratio*

$$af(N(u), N(A)) = \frac{|N(u) \cap N(A)|}{|N(u)|} \tag{1}$$

If $N(u) = \emptyset$ then af(N(u),N(A))=0. The best new node $b \in N(A) \setminus A$ is the node which maximises this affinity score [1].

Here it depends which seed topic is taken as the first one. This algorithm may find for particular topic A some other node B as the best node. But this does not have to find topic A as the best node for topic B if B is a seed topic.

3 Methodology

This chapter describes our approach to retrieve topics from text and how to construct topics hierarchy. Source of our data are manually extracted aims and scopes from DBLP journals. Follows automatic extraction particular topics from these texts.

3.1 The algorithm for creating of dictionary

Let R be the set of abstracts where $r_i \in R$ is abstract of journal. Then let N be the set of improper words (negative dictionary), where $n_m \in T$ is improper word in negative dictionary. Also we define δ, that represents maximal number of words in topic. For example if $\delta = 3$ then topic contantains max three words. Function *split* is standart function of programing language C# and splits text by conjuction.

Algorithm is based on searching of specific important conjunctions in sentences (and, or, comma, semi-comma) and words with a short distance to these conjunctions. These conjunctions have various priority whereas the most important is *and*. Other special characters were removed. In the case of *and* conjunction is necessary to correctly analyze the topic which we do following way:

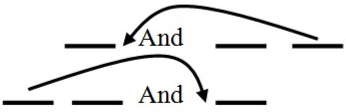

Fig. 1. Dividing of topic containing *and*

In fig.1 is an example of syntactic compound of topic with *and.* Arrows show which word is needed to add and where. For instance *software and hardware architecture* or other example *software architecture and hardware.* In the fig.1 is suggested an algorithm for topics consisting of two words. Similar algorithm is for 3-word topics. It is important to proceed from lexical structure of more-words conjunctions when creating such rules.

It might happen that this method divides the topic into two new topics which were not supposed to be divided, e.g. *Data terminals and printers.* Other conjunctions usually does not have this issue and these conjunctions have for semantic of topics also the same priority. Next step is to save these parsed topics into incident matrix which creates relations between particular nodes. Then we can easily set up level of importance according to frequency of their occurrence in text.

Algorithm 1 The algorithm for create dictionary

```
Require: $\delta = 3$, R=set of abstracts, N=set of improper words
  for all $r_i \in R$ do
    S=split($r_i$, '.') {where S is the set of sentences for $r_i \in R$}
    for all $s_j \in S$ do
      W=split($s_j$,conjunction) {where W is potencial the set of topics for $s_j \in S$}
      for all $w_k \in W$ do
        if $|w_k| <= \delta$ and $w_k \notin N$ then
          Add $w_k$ to T {where T is the set of topics}
        end if
      end for
    end for
  end for
```

3.2 The algorithm for create graph of topics

Let T be the set of topics, where $t_i \in T$ is topic in dictionary of topics. Let R be the set of abstracts, where $r_i \in R$ is abstract of journal. Also we use incidence matrix $I^{m \times n}$, that represents link between words in dictionary.We define ϵ that represents required strength of relationship between two words in incidence matrix.

In the first phase we analyze each journal separately and we search for one, two or three-word topics.

4 Testing

We identified about 750 journals in DBLP.For testing we used 500 aims and scopes from these journals and containing about 90 000 words. This text was parsed by algorithm 1 into about 4000 topics. During testing we found out that

Algorithm 2 The algorithm for searching of similar topics

```
Require: $I^{m \times n}$=empty matrix, $m = n = |T|$, $\epsilon$=required strength of relationship,
  A=the set of seed topics, x=number of topics
  for all $r_i \in R$ and do
    S=split($r_i$, '.') {where S is the set of sentences for $r_i \in R$}
    for all $s_j \in S$ do
      W=split($s_j$,conjunction) {where W is potencial the set of topics for $s_j \in S$}
      Compute $Z = W \cap T$ {where Z is the set of topics in sentence}
      for all $z_n, z_m \in Z$ do
        Create link between $z_m, z_n$ in I, where $i_{z_m z_n} = i_{z_m z_n} + 1$.
      end for
    end for
  end for
  for all $i_{mn} \in I$ do
    if $i_{mn} < \epsilon$ then
      $i_{mn} = 0$
    end if
  end for
  while $|A| < x$ do
    Compute $N(A)$ from matrix I
    for all $b_i \in N(A) \setminus A$ do
      Compute $N(b_i)$ from matrix I
      Compute $af(N(b_i), N(A))$
    end for
    Add $b_i$ with the highest affinity to A.($b_i \cup A$)
  end while
```

for our topic-database size we are obtaining the best results if we take first 1000 topics with the highest frequency of occurrence. So there is about 3000 topics left but their rating is too low to affect the result significantly.

We achieved this result by following way: first we took 400 topics, but in this setting we found only few related topics. Then why we doubled size of topics. In this way we continued up to 2000 topics and we were observing generated groups. We were also focusing on reliability of topics in particular groups so we do not obtain topics out of seed topic in group. We set a border up to 10 good topics in each group, if possible. However for some of topics algorithm finished in lower number. From all results we found that for our database is optimal to take first 1000 topics with the highest number of occurrences. We compared our results to ACM hierarchy.

5 Conlusion

Most from previous works were using WordNet database for testing. Because topics are semantically different we could not use this one but instead we used ACM hierarchy database which is a database of topics hierarchically ordered. It is possible to download also in XML so it is very easy to work with it. Even we

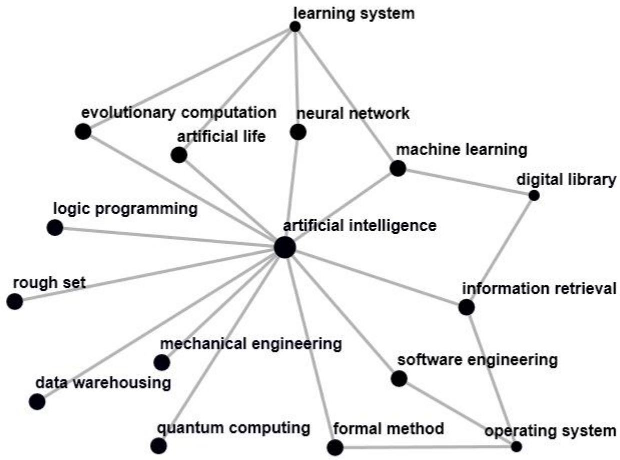

Fig. 2. Graph generated for topic *Artificial Intelligence*

are not able to measure accuracy as a single number, we can pursue conformity to ACM.

Result is strongly affected by the age of last release of ACM hierarchy (in 1998) when many topics did not exist, for instance social networks or business intelligence. Our algorithm found about 5000 automatically generated topics from 500 journals. There were manually found approximately 10% of topics which were not good topics. Latest ACM database consists of 1200 topics. We choose some topics common for ACM and our results and compared them. Since we found only about 20 percent of topics same in both databases it does not make much sence to compare them together. The difference between both databases is that for instance in ACM there is a single topic *graphic* but our algorithm generates *computer graphic* and 8 other topics containing word *graphic*. This is also a reason why is our hierarchy so much larger.

Figure 4 shows how algorithm is adding every new topic into hierarchy. First are added topics as first level which were directly connected to the seed topic, then to the level 2 were added topics with the highest ratio to topics from first level. This algorithm iterates the same for next levels.

In Table 1 we present results for few randomly chosen seed topics. We can see that our method generates corectly related topics to the seed topic. However there might be some topics not related to the seed word and these topics are marked in table as italic font. Compare to ACM we obtained much better related topics in groups. More aims and scopes we have better results we get.

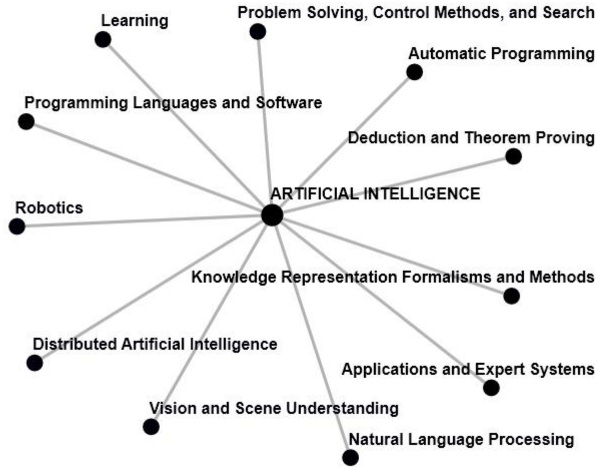
Learning
Problem Solving, Control Methods, and Search
Automatic Programming
Programming Languages and Software
Deduction and Theorem Proving
ARTIFICIAL INTELLIGENCE
Robotics
Knowledge Representation Formalisms and Methods
Distributed Artificial Intelligence
Applications and Expert Systems
Vision and Scene Understanding
Natural Language Processing

Fig. 3. ACM graph for topic *Artificial Intelligence*

References

1. Widdows D., Beate Dorow: A Graph Model for Unsupervised Lexical Acquisition. *COLING, 2002.*
2. Widdows D. and Ferraro K.: Semantic vectors: A scalable open source package and online technology management application. *In Proceedings of the sixth international conference on Language Resources and Evaluation,2008.*
3. Widdows, D.: Orthogonal negation in vector spaces for modeling word-meanings and document retrieval. *In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics,2003.*
4. Rieffel, E.: Certainty and uncertainly in quantum information processing. *In Bruza et al. (2007)*, 134-141.
5. Widdows, D. and Bruza P.: Quantum information dynamics and open world science. *In Bruza et al. (2007)*,126-133.
6. Ellen Riloff and Rosie Jones: Learning dictionaries for information extraction by multi-level bootstrapping. *In Proceedings of the Sixteenth National Conference on Artificial Intelligence, 1999*
7. Brian Roark and Eugene Charniak: Noun-phrase co-occurence statistics for semi-automatic semantic lexicon construction. *In COLING-ACL,1998*
8. Trevor P. Martin, Masrah Azmi-Murad: An Incremental Algorithm to find Asymmetric Word Similarities for Fuzzy Text Mining. *WSTST 2005*, 838-847
9. Communications of the ACM, New York, NY, USA
10. Dekang Lin: Automatic retrieval and clustering similar words,*In COLINGACL,Montreal,Athens*

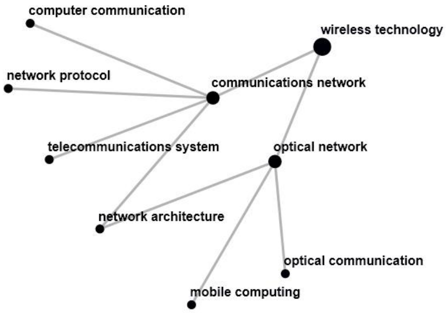

Fig. 4. Hierarchical graph generated for *Wireless technology*

Table 1. Groups of topics for chosen seed topics

Seed topic	Topics
artificial intelligence	formation retrieval, *digital library*, *mechanical engineering*, *data warehousing*, logic programming, machine learning, evolutionary computation, quantum computing, artificial life, learning system, neural network
wireless	fspread-spectrum system, cellular system, *ip*, adaptive antenna, network protocol, telecommunications network, computer communication, *lan*, *man*, satellite network, wireless computing
computer graphic	image processing, computer vision, *speech recognition*, data mining, machine learning, digital library, evolutionary computation, artificial life, learning system, data warehousing
mathematical method	computational method, numerical mathematic, computational mathematic, stochastic process, set theory, probability theory, matroid theory, computer scientist, coding theory, differential equation,
process engineering	production engineering, electronics engineering, unmanned system, electrical engineering, chemical engineering, mechanical engineering, traffic engineering, *applied mathematic*
language research	language technology, human-computer interaction, decision support, computational linguistic, spatial reasoning, qualitative reasoning, *global warming*, semantic web, object-oriented programming
combinatorial optimization	randomized algorithm, mathematical programming, graph algorithm, indexing information, data analysis, *computer vision*, information fusion, *database design*, *web mining*, risk analysis