COULD WE AUTOMATICALLY REPRODUCE SEMANTIC RELATIONS OF AN INFORMATION RETRIEVAL THESAURUS?

Alexander Panchenko
Center for Natural Language Processing (CENTAL),
Université catholique de Louvain
e-mail: alexander.panchenko@student.uclouvain.be

Abstract

A well constructed thesaurus is recognized as a valuable source of semantic information for various applications, especially for Information Retrieval. The main hindrances to using thesaurus-oriented approaches are the high complexity and cost of manual thesauri creation. This paper addresses the problem of automatic thesaurus construction, namely we study the quality of automatically extracted semantic relations as compared with the semantic relations of a manually crafted thesaurus. The vector-space model based on syntactic contexts was used to reproduce relations between the terms of a manually constructed thesaurus. We propose a simple algorithm for representing both single word and multiword terms in the distributional space of syntactic contexts. Furthermore, we propose a method for evaluation quality of the extracted relations. Our experiments show significant difference between the automatically and manually constructed relations: while many of the automatically generated relations are relevant, just a small part of them could be found in the original thesaurus.

Keywords: *thesaurus, semantic relations, vector-space model, distributional analysis, multiword expressions.*

1. INTRODUCTION

An information retrieval thesaurus describes a certain knowledge domain by listing all its main concepts and semantic relations between them. In their simplest form thesauri consist of a list of important terms and semantic relations between them (see Figure 1). Thesauri have been used in documentation management projects for years. They were even used by libraries and documentation centers long before the computer era. This long tradition and the more recent success of the thesaurus based information systems has led to adoption of thesaurus-based techniques by the industry and to the development of international standards1.

According to Foskett [1], the main purposes to use a thesaurus are (1) to provide a standard vocabulary for indexing and searching, (2) to assist users with locating terms for proper query formulation, and (3) to provide classified hierarchies that allow the broadening and narrowing of the current request according to the needs of the user.

EuroVOC [2] is one example of a big contemporary information retrieval thesaurus: it is used for indexing documents of the European

energy industry
- **NT1 energy conversion**
  - RT soft energy (6626)
- **NT1 energy technology**
  - RT bioconversion (6411)
  - RT oil technology (6616)
  - RT soft energy (6626)
- **NT2 fuel cell**
- **NT1 energy-generating product**
- **NT1 fuel**
  - RT energy resources (5211)
- **NT2 fossil fuel**
  - RT coal (6611)
  - RT natural gas (6616)
  - RT petroleum (6616)

Figure 1. A term with relations (EuroVOC)

Parliament, the Office for Official Publications of the European Communities, and many other European institutions. Another well-known thesaurus is AgroVOC [3] — a multilingual, structured and controlled vocabulary designed to cover the terminology of all subject fields in agriculture, forestry, fisheries, food and related domains. This resource was created by the Food and Agriculture Organization of the United Nations (FAO) and has many applications all over the world.

Apart from the applications in Information Retrieval [4], the semantic information contained in thesauri and ontologies was used in solving technical problems such as Text Categorization [5], Term Extraction [6], developing Question Answering systems [7] and some others.

The traditional way of thesaurus construction involves great amount of manual labor and proved to be very time consuming and costly. Furthermore, it does not allow for an easy way to keep semantic resources updated. All these factors limit applications of thesaurus-oriented approaches. One of the solutions to this problem is to automatize thesaurus construction, as it was proposed for instance in our previous work [8]. Basically, the automatized procces comprises two main steps: selecting key terms for a given domain and establishing semantic relations such as synonymy, hyponymy, and association between them. Important question concerns the quality of an automatically generated thesaurus. In this paper we investigate how similar are the automatically generated semantic relations and the semantic relations established by an expert. In our experiments we use vocabulary of a manually constructed thesaurus and try to reconstruct semantic relations between its terms by means of distributional analysis.

The paper is organized as follows. The section 2 lists some related research. We present our dataset in the 3rd section. The section 4 gives description of our method for mining semantic relations from corpus and from §4.1 to §4.4 we give details about each of its steps. Then, in section 5, we present our approach for evaluation set of automatically constructed relations and its results for our dataset. We show that while many of the automatically extracted relations make sense, the model did not recall many of the manually crafted relations. Finally we sum up the main points of this paper in section 6.

2. RELATED WORK

There has been proposed number of approaches for automatic discovering of semantic relations between words: with help of lexical and dependency patterns [9], based on Latent Semantic Analysis [10], from evidence contained in electronic dictionaries [11] or encyclopedias [12], and even from the Web link structure [13].

Yet another well-known method for discovering semantic relations between terms relies on the Distributional Hypothesis of Harris [14] which states that "words that occur in the same contexts tend to have similar meanings". Schutze [15] proposed to represent word as a vector in a multidimensional space of all possible contexts. The spatial proximity between terms in this model indicates how similar their meanings are. There have been proposed different variations of this thesaurus construction method (e.g. [16], [17], [18] or [19]), especially in combination with clustering techniques such as in the work of Sharon [19] or Pantel and Lin [20]. We use the vector-space model based on syntactic contexts as in the work of Grefenstette [21], and extend it to deal also with multiword expressions and not only with nouns as in the original work.

3. DATASET

The dataset we are working with comprises two parts: a 20 million word corpus of political texts in French and a manually constructed thesaurus. The corpus comprises 11.386 text documents coming from a governmental institution, such as deputies' requests to ministers, protocols of parliamentary sessions, international conventions, activity reports, texts of propositions of new laws and so on.

The thesaurus was constructed manually based on the analysis of the described above corpus. The semantic resource aims to provide vocabulary for indexing documents of a governmental institution such as a parliament, thus it comprises different terms coming from various domains (12 in

our case) which are often discussed in such an institution e.g. legislation, economics, finances, international relations etc. The thesaurus contains n=2514 concepts $C=\{c_1,...,c_n\}$ where every concept c_i is represented with j terms $\{d_{i1},...,d_{ij}\}$ which are synonyms or quasi-synonyms. For example, the concept "Aircraft" is composed of eight terms (here and in further examples provide the corresponding translation from French for convenience of the reader): $c_i=\{d_{i1},...,d_{i8}\}$={Aircraft, Airship, Plane, Aerostat, Helicopter,..., Dirigeable}.

The terms are the key part of the thesaurus — its vocabulary, they reflect main concepts of a certain domain. The vocabulary of the thesaurus D comprises m=4771 terms:

$$D = \bigcup_{c_i \in C} c_i = \{d_1, ..., d_m\}.$$

Most of the terms in the vocabulary (65%) are noun phrases, such as "ultra-lightweight aircraft" or "hot-air balloon", and the rest 35% of terms are nouns, like "airplane" or "aerostat". The concepts are organized in the hierarchy with set of 2456 hyponymy relations R^{NT}. Furthermore, the concepts of the thesaurus are interconnected with the set of 1530 associative relations R^{RT}. Every semantic relation $r_{ij} \in \{R^{NT} \cup R^{RT}\}$ defines a semantic link between concepts c_i and c_j represented by the ordered pair $<c_i,c_j>$. Thus, the thesaurus is the oriented graph (network) $T=(C,R)$ having the concepts of the thesaurus C as nodes, and the semantic relations between concepts $R=R^{NT} \cup R^{RT}$ as edges.

4. CONSTRUCTING SEMANTIC RELATIONS BETWEEN CONCEPTS

Given a corpus and a set of concepts or terms, the goal of our method is to construct semantic relations between them. We use the distributional analysis [21] to construct set of semantic relations between terms of the original thesaurus. In this model every input concept is modeled as a point in the distributional space of all possible syntactic contexts. The procedure of calculating relations between the concepts involves preprocessing, indexing terms, constructing distributional space of terms, and calculation of relations between terms. The following paragraphs describe the respective steps of the proposed method.

4.1. Preprocessing vocabulary and corpus

The goal of the first step is to perform cleansing of the dataset: we use regular expressions to normalize whitespaces, remove corrupted character sequences, and some meta-information, such as document identifiers, from the texts. Also at this step we deaccent documents and terms by substituting the characters with French diacritic symbols such as "à" or "é" with their non accented equivalents.

4.2. Indexing terms

The goal of this step is to find all occurrences of the terms $d \in D$ in the corpus and save information about their positions in some index. In order to deal with linguistic variation and some typos we search terms with help of regular expressions. We use the Algorithm 1 to generate a regular expression for each term of the thesaurus. The procedure relies on the stemming function **Stem**() (we use a simplified version of the Porter stemming algorithm, which strips endings like «s», «es», and «aux» for long words) and the function **GetType**() (the function use stop-lists and regular expressions. The type «articles or prepositions» was defined with the 28 function words: de, du, la, le, les, des, d', l', d, l, a, aux, et, au, en, pour, dans, par, car, dont, donc, comme, que, plus, encore, entre, vers, via) which returns type of an

```
Input: Descriptor d
Output: Regular expression re for searching descriptor d in text
re ← "" ;
foreach word ∈ d do
    switch GetType(word) do
        case article or preposition
            re ← re + "((de|du|la...|vers|via)\s+|)";
        case regular word
            rword ← Stem(word) + "\w{0,3}";
            Replace(rword,"-","(-|\s+|)") ;
            re ← re + rword ;
        case abbreviation
            aword ← word without dots and spaces ;
            if GetType(aword) ≠ stopword then
                spacer ← "(\.|\s+|)";
            else
                spacer ← "(\.|\s+)";
            foreach letter ∈ word do
                re ← re + letter + spacer ;
        otherwise
            re ← re + word ;
    if word is not the last one then
        re ← re+ "\s+";
return "\b" +re+ "\b"
```

Algorithm 1. Calculating regular expression for a descriptor

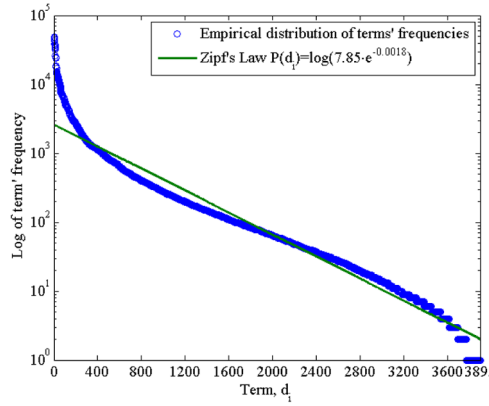

Figure 2. Empirical distribution of thesaurus term frequences compared with the Zipf's law

input word. The Algorithm 1 replaces every word of an input term with a regular expression pattern. The procedure replaces every article or preposition with the conjunction of several articles and prepositions (line 4–5). A regular word is replaced by regular expression based on word's stem form (lines 6–9). Finally, special spacer inserted after every letter of an abbreviation word (see lines 10–17). The described procedure will transform the term "conventions internationales" (international conventions) the following regular expression:

```
\bconvention\w{0,3}\s+internationale\w{0,3}\b
```

This regular expression captures both singular form "convention internationale" and plural form "conventions internationals" of the phrase. Similarly, the automatically generated regular expression for the term "modification de la legislation" (modification of legislation) will capture different pertinent variations of this term such as "modifications de la legislation", "modification a la legislation", or "modifications dans la legislation".

We run the Algorithm 1 for every term d of the thesaurus and save information about every term occurrence in the index record $<d,doc,p^{beg},p^{end}>$, where p^{beg} and p^{end} are positions of the beginning and the end of the term in the document *doc*. Set of all index records compose the index I.

The Figure 2 shows that the terms' frequency distribution approximately follows the Zipf's Law [22]. Although, one can see that the real distribution doesn't ideally fit the Zipf's distribution in the area of very high- and low-frequency terms. It is mostly due to the fact that our vocabulary is just a subset of the real vocabulary of the corpus.

4.3. Constructing distributional space of terms

To construct the *distributional space* associated to the corpus we use syntactic dependencies between words of sentences where at least one term $d \in D$ was found. In our experiments we used XIP natural language parser [23] to produce set of syntactic dependencies *SR* from the corpus. Every dependency $<w_1,p_1^{beg},t,w_2,p_2^{beg}>$ contains information about the syntactic relation of type t between the word w_1 starting at the position p_1^{beg} and the word w_2 starting at the position p_2^{beg}. Some syntactic relations such as dependency between a nominal head and a determiner (e.g. <the,0,DET,helicopter,5>) brings little information about the semantics of the head word. We choose 9

Table 1. Syntactic relations used to construct distributional space by A) the author B) Piersman et al. [24] C) Hindle [25], D) Hirshman et al. [26], E) Hatzivassiloglou et al. [27], F) Lonneke [28], G) Takenobu et al. [29], F) Grefenstette [21]

Acronym	Description of syntactic relation*	A	B	C	D	E	F	G	F
ADJMOD	Attaches the modifier of adjective to the adjective itself.	**X**	X		X	X		X	X
CONNECT	Links the verb of a finite clause to the grammatical word that introduces the clause.	**X**				X	X		
COORD	Coordination. This binary relation links coordinated elements.	**X**	X			X		X	
DOBJ	This dependency attaches a deep object to the verb.	**X**				X	X		
DSUBJ	This dependency attaches a deep subject to the verb.	**X**				X	X		
NMOD	Attaches a modifier to the noun it modifies.	**X**				X			X
OBJ	Attaches a direct object to its verb.	**X**	X	X	X	X	X	X	X
SUBJ	Attaches the surface subject to the verb, including infinitive verbs.	**X**	X	X	X	X	X	X	X
VMOD	Attaches a modifier of a verb to the verb itself.	**X**				X	X		
DET	Links a nominal head and a determiner.				X	X			
APP	Apposition. Links two adjacent units that have identical referents.					X		X	X
PREPOBJ	Attaches a preposition to the noun or the verb it precedes.		X			X		X	X

* we adopted these descriptions mostly from the documentation of the XIP parser [23].

syntactic relations listed in Table 1 to construct the distributional space of terms. The table also indicates what syntactic relations were used in experiments of some other researchers. This comparison is not exhaustive, but still we can observe that the most popular relations are the OBJ, SUBJ, and ADJMOD. One can assume that these types of syntactic relations provide the best clues about meaning of a term.

At this stage we have to define a distributional space and represent the terms of thesaurus in this space. The dimensions of the distributional space must be such that they let us distinguish terms with different meanings. In our approach the dimensions of the n-dimensional distributional space are associated with the *syntactic contexts* $B=\{\beta_1,...,\beta_n\}$. Every syntactic context is a tuple $<t,w>$ composed of the lemmatized word w and the type of syntactic relation t. We derive set of syntactic contexts (features) from the set of extracted syntactic dependencies SR. Basically, one tuple $<w_1,p_1^{beg},t,w_2,p_2^{beg}>$ gives two syntactic contexts $<t,w_1>$ and $<t,w_2>$. Every term d_i is represented with a vector $\mathbf{f}_i$ in the distributional space. The feature matrix $\mathbf{F}=(\mathbf{f}_1,...,\mathbf{f}_m)^T$ has m rows and n columns, the i-th row of this matrix corresponds to the term d_i and j-th column corresponds to the syntactic feature β_j.

We use the Algorithm 2 to calculate the dimensions of the distributional space B and the feature matrix $\mathbf{F}$. The majority of the previous algorithms represent a single word or chunk in the distributional space (e.g. [21], [24], or [29]). The main difference of our algorithm is what it can calculate distributional representation of an arbitrary multiword expression. Basically, it calculates the distributional representation of a term as a sum of syntactic contexts of all its non-stopwords, excluding dependencies with stopwords and words inside the term (see Figure 3). The algorithm takes as input set syntactic dependencies SR, index I containing positions of all occurrences of terms in the corpus, and the stoplists. At the first step the algorithm creates void set of syntactic contexts B and void multiset C. An element of the multiset C is a tuple $<d,\beta>$ which maps a term d and a

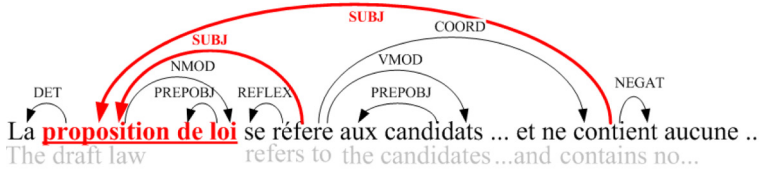

Figure 3. Syntactic dependencies, extracted from the text and syntactic contexts of the term “proposition de loi”

Input: Syntactic dependencies $SR = \{SR_1, ..., SR_K\}$ for K documents, terms D, term index I, stop parts-**o**f-speech SP, stopwords SW, allowed types of syntactic dependencies T, syntactic context threshold β^T

Output: $B-$ distributional space, $\mathbf{F}-$ feature matrix

1 $C \leftarrow \emptyset, B \leftarrow \emptyset, w_{context} \leftarrow$ "" , $w_{term} \leftarrow$ "" ;

// Calculating set of syntactic features B and multiset C

2 **foreach** *document* k *in corpus* **do**

3 **foreach** $\left\langle w_i, p_i^{beg}, t, w_j, p_j^{beg} \right\rangle \in SR_k$ **do**

4 **if** $\exists \langle k, d, p^{beg}, p^{end} \rangle \in I : p_i^{beg} \in [p^{beg}; p^{end}]$ **then**

5 $w_{context} \leftarrow w_i$;

6 $w_{term} \leftarrow d$;

7 **else if** $\exists \langle k, d, p^{beg}, p^{end} \rangle \in I : p_j^{beg} \in [p^{beg}; p^{end}]$ **then**

8 $w_{context} \leftarrow w_j$;

9 $w_{term} \leftarrow d$;

10 **else**

11 continue ;

12 **if** $w_{context} \notin w_{term}$ **and** $w_{term} \notin w_{context}$ **and** $t \in T$ **and**

13 GetPOS($w_{context}$) $\notin SP$ **and** $w_{context} \notin SW$ **then**

14 $\beta \leftarrow \langle t_j, w_{context} \rangle$;

15 $B \leftarrow B \cup \beta$;

16 $C \leftarrow C \cup \langle w_{term}, \beta \rangle$;

// Calculating feature matrix $\mathbf{F}$

17 $\mathbf{F} \leftarrow \mathbf{0}_{|D|,|B|}$;

18 **foreach** $\langle d_i, \beta_j \rangle \in C$ **do**

19 $f_{ij} \leftarrow f_{ij} + 1$;

20 Normalize($\mathbf{F}$) ;

21 GroupContexts($\mathbf{F}, B$) ;

22 RemoveContexts($\mathbf{F}, B, \beta^T$) ;

23 **return** $B, \mathbf{F}$

Algorithm 2. Calculation of feature space *B* and feature matrix **F**

syntactic context β. Then the algorithm incrementally fills these two sets by checking every extracted syntactic tuple (lines 2–16). In particular, if the word w_1 from the dependency $<w_1,p_1^{beg},t,w_2,p_2^{beg}>$ belongs to the term d then we add the syntactic context $<t,w_2>$ to the term d. Similarly, if the term index I contains a record indicating that the word w_2 belongs to the term d we add new syntactic context $<t,w_1>$ to the d. Furthermore, the algorithm will not add the syntactic context $<t,w_{context}>$ to the term d if the context word $w_{context}$ is a part of term d, or if it is a stopword (lines 12–13).

The second part of the algorithm (lines 17–21) constructs the feature matrix **F** from the multiset *C*. Firstly, we set every element f_{ij} of this matrix equal to the number of times term d_i occurred with the context β_j (lines 18-19). Then, we normalize the feature matrix as follows (line 20):

$$f'_{ij} = \frac{f_{ij}}{|d_i| \cdot |\beta_j|}.$$

In the previous formula $|d_i|$ is the number of times the term d_i occurred in the corpus and $|\beta_j|$ is the number of times the syntactic context β_j occurred in the corpus. After the normalization every element of the feature matrix belong interval between zero and one: $f_{ij} \in [0;1]$.

The procedure **GroupContexts**() reduces sparsity of the distributional space by merging the similar syntactic contexts such as <*NMOD*,37 millions> and <*NMOD*,71 millions>. The procedure groups features representing dates, sums of money, ordinal numbers, real numbers and percents. Finally, the procedure **RemoveContexts**()deletes the syntactic contexts which occurred less than β^T times in the corpus: $B' = \{\beta \in B : |\beta| \geq \beta^T\}$. We present results of experiments with different values of this parameter in the section 5.2.

4.2. Calculations of relations between terms

We calculate measures of semantic similarity between terms d_i and d_j with cosine between their respective vectors

$$sim(d_i, d_j) = s_{ij} = \frac{\mathbf{f}_i \cdot \mathbf{f}_j}{\|\mathbf{f}_i\| \cdot \|\mathbf{f}_j\|}.$$

We define set of related terms for the term *d* as the set of its nearest neighbors. We calculate set of relations between terms by thesholding the similarity matrix **S** with the threshold s^T: $\hat{R} = \{\langle t_i, t_j \rangle : s_{ij} \geq s^T\}$.

5. EVALUATION

5.1. Assessment protocol

Our evaluation is based on the idea that among all possible automatically constructed thesauri $\{(C, \hat{R}_1), (C, \hat{R}_2), ...\}$ the best one is the one which is the most similar to the manually constructed thesaurus $T=(C,R)$. We evaluate quality of the automatically constructed relations with the exact and the fuzzy precision measures. The exact precision measure is defined as number of automatically extracted relations which are found in the manually constructed thesaurus, divided by the total number of extracted relations:

$$precision^E = \frac{|\hat{R} \cap R|}{|\hat{R}|}.$$

The original thesaurus is a hand crafted linguistic resource containing 3986 different semantic relations between 2514 concepts. It was created by a concrete group of experts, and if another group of experts would be asked to build the same thesaurus they would created a different semantic resource. Therefore the thesaurus contains not exhaustive list of semantic links between the concepts, and the exact precision measure could tend to underestimate the real precision rate. Let us illustrate this issue on the following example: in one of our experiments the algorithm discovered that the term "foreign public act" is related to the three following terms "private international law", "civil procedure", "arbitration". Meanwhile, the original thesaurus contains two different terms related to the "foreign public act": "legal act" and "foreign legislation". There is no overlap between these lists of related terms, thus the exact precision rate will equal zero. Normally, we would like to deal with more flexible evaluation measure.

We propose the *fuzzy precision* measure which addresses this problem by taking into account short paths between terms into the original thesaurus. Indeed, we found that the thesaurus contains the following short transit paths between the term "foreign public act" and the automatically discovered terms:

foreign public act → foreign legislation → branch of law → private international law
foreign public act → legal act → course of law → civil procedure
foreign public act → legal act → course of law → civil procedure → arbitration

To calculate the fuzzy precision score we generate set of *fuzzy semantic relations* R^{Fk} and use it as a golden standard for evaluating quality of the automatically constructed relations. Generating set of fuzzy relations comprises the three following steps:

1. Constructing adjacency **W** matrix of the thesaurus T defined as follows:

$$w_{ij} = \begin{cases} 2 & \text{if } \exists\langle d_i, d_j\rangle \in R^{NT} \\ 1 \text{ if } (\exists\langle d_i, d_j\rangle \in R^{NT}) \vee (\exists\langle d_i, d_j\rangle \in R^{RT}) \vee (\exists\langle d_j, d_i\rangle \in R^{NT}) \\ 0 & \text{otherwise} \end{cases}$$

2. Calculating matrix of shortest paths **P** between concepts of the thesaurus T with the Floyd's algorithm [30]. An element of this matrix p_{ij} contains length of the shortest path between the concepts c_i and c_j.

3. Calculating set of *fuzzy relations* R^{Fk} between terms. This set contains pairs of terms connected by a path in the original thesaurus with length less or equal than k: $R^{Fk}=\{<c_i,c_j>:p_{ij}\leq k\}$.

In our experiments we constructed two fuzzy versions of the original thesaurus: R^{F3} and R^{F4}. The first set contained 80.641 pairs of concepts linked by a path in the thesaurus with length less or equal than k=3.

The second set contained 254.441 relations; it was constructed with the maximum path length equals to k=4. The fuzzy precision measure is defined as number of automatically extracted relations which were found in the corresponding version of the fuzzy thesaurus, divided by the total number of extracted relations:

$$precision^{Fk} = \frac{\left|\hat{R} \cap R^{Fk}\right|}{\left|\hat{R}\right|}, k = \{3,4\}.$$

5.2. Results

The Table 2 presents some relations between terms of the thesaurus which were automatically extracted from the corpus with the described method. The number in brackets is the length of the shortest path in the original thesaurus T between the term from the left column and the term from the right column.

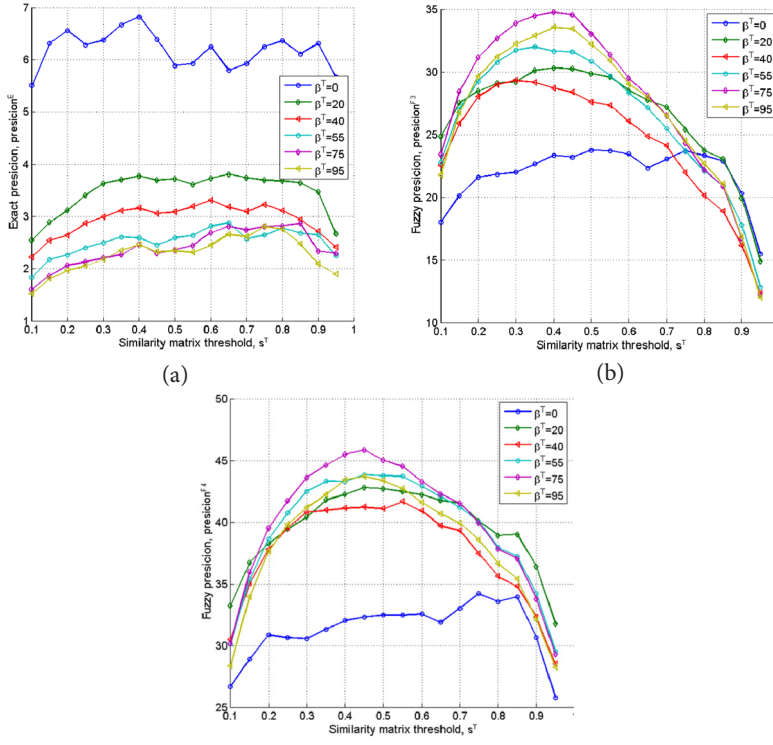

Figure 4. (a) Exact precision measure, (b) Fuzzy precision measure k=3, (c) Fuzzy precision measure k=4

Table 2. Comparison of automatically and manually constructed relations between terms of the thesaurus (we used the following parameters to generate these relations: s^T=0.4, β^T=75)

Term	Manually constructed	Automatically generated
administration of taxes	administration of the state	administration of the cadastre and the topography (2), state socio-educational center (8), public education (4), cultural institution (8), institute of hygiene and public health (7), state vineyard station (6)
admission to studies	school organization, education, admission to employment	archives of the state (9), certificate of teacher (6), program of studies (2)
medical assistance	medical organization	emergency medical services (1), medical analysis (6), medically assisted procreation (6) hygiene (6), wine institute (9), medical organization (1) medical profession (3), vaccination (5)
european election	election, political life, european parliament	legislative election (2)
unemployed person	unemployment, employment, employment administration	unemployment compensation (2)
education grants	school life, education	youth movement (11)
european community	european organisation, single european act, yaounde agreement, lome convention	european defense community (1), european atomic energy community (1), european coal and steel community (1), international economic partnership (2), country union (2)
school leaving certificate	diploma, promotion of students, school environment	foreign education certificate (2)
maternity leave	leave, number of hours, work	parental leave (3), work schedule (3)
south africa	foreign country	saudi arabia (2), bahamas (2), belize (2), colombia (2) comoros (2) congo (2), djibouti (2), united arab emirates (2), eritrea (2), federated states of micronesia (2), mexico (2), gabon (2), guinea (2), equatorial guinea (2), guyana (2), kazakhstan (2)

We conducted several experiments with different values of the minimum syntactic context frequency $\beta^T \in [0;\infty]$ and the similarity matrix threshold $s^T \in [0;1]$. The figure 4(a) shows that the automatically and manually constructed relations are completely different with respect to the exact quality measure $precision^E$: the highest value of this rate is around 7%. This rate was obtained by the model keeping all the syntactic features ($\beta^T=0$) and with similarity threshold value $s^T=0.4$.

The figures 4(b) and 4(c) show that for $k=4$ roughly every second (every third for $k=3$) automatically extracted relation is present in the original thesaurus: the highest values of the fuzzy precision measure are $precision^{F4}=46\%$ and $precision^{F3}=35\%$, respectively. These scores were achieved also with the similarity matrix threshold $s^T=0.4$, but on the distributional space composed of the syntactic contexts occurred more than 75 times in corpus: $\beta^T=75$.

6. CONCLUSION AND FUTURE WORK

Firstly, we proposed a simple method for extracting semantic relations between multiword terms based on the distributional analysis. The method was used to reproduce semantic relations between terms of the manually constructed Information Retrieval thesaurus. Secondly, we proposed a technique for evaluating the quality of the automatically extracted relations based on fuzzy versions of the manually constructed thesaurus.

The answer to the question in the title of the article is as follows: the proposed method cannot exactly reproduce relations from the original thesaurus, but it is capable of finding pairs of terms linked with a short path in the original thesaurus. The experiments show significant difference between the automatically and manually constructed relations. Nevertheless, our observations suggest that the proposed method can discover new relevant relations between terms. We conclude that the method could be useful in the process of automatic thesaurus construction, but its results might require moderation of an expert.

The future work will be focused on overcoming the main limitations of the method: low precision rate, need to tune the threshold parameters, and the fact that the method does not return type of the extracted relations.

7. ACKNOWLEDGMENTS

The author wishes to thank Wallonie-Bruxelles International organisation for supporting this research and professors Cédrick Fairon, Yuri Philippovich and Marco Saerens for their discerning comments and notable suggestions about this work. The author also thanks Xerox Research Centre Europe for providing their parser for French.

REFERENCES

1. **Foskett D. J.**, Readings in information retrieval.: Morgan Kaufmann Multimedia Information And Systems Series, San Francisco, CA, USA, 1997, pp. 111–134.
2. EuroVOC. Office for Official Publications of the European Communities. [Online]. http://europa.eu/eurovoc/
3. AgroVOC. Food and Agriculture Organization of the United Nations (FAO). [Online]. http://www.fao.org/agrovoc/
4. **Alan R. Aronson, Rindflesch C. Thomas, and Browne C. Allen.**, «Exploiting a large thesaurus for information retrieval,» in Proceedings of RIAO, 1994, pp. 197–216.
5. **Bang S., Yang J., and Yang H.**, «Hierarchical document categorization with k-NN and concept-based thesauri,» Information Processing and Management: an International Journal, Volume 42 , Issue 2, pp. 387–406, 2006.
6. **Taketoshi Yoshidaa, and Xijin Tang Wen Zhanga**, «Using ontology to improve precision of terminology extraction from documents ,» Expert Systems with Applications, vol. 36, no. 5, pp. 9333–9339, 2009.
7. **Gideon Mann**, «Fine-grained proper noun ontologies for question answering,» in International Conference On Computational Linguistics,COLING-02 on SEMANET: building and using semantic networks - Volume 11, 2002, pp. 1–7.
8. **A. Panchenko**, «Technology of the automated thesaurus construction for Information Retrieval,» Intelligence Systems and Technologies, Bauman Moscow State Technical University, Moscow, vol. 9, pp. 124–140, 2009.
9. **Erik Tjong Kim Sang and Katja Hofmann**, «Lexical patterns or dependency patterns: which is better for hypernym extraction?,» in Proceedings of the Thirteenth Conference on Computational Natural Language Learning, 2009, pp. 174–182.
10. **Tonio Wandmacher**, «How semantic is Latent Semantic Analysis?,» in in Proceedings of TALN/RECITAL, 2005, pp. 6–10.
11. **Edward A. Fox, J. Terry Nutter, Thomas Ahlswede, Martha Evens, and Judith Markowitz**, «Building a large thesaurus for information retrieval,» in Proceedings of the second conference on Applied natural language processing, 1988, pp. 101–108.
12. **Milne David, Medelyan Olena, and Witten Ian H.**, «Mining Domain-Specific Thesauri from Wikipedia: A Case Study,» in Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, 2006, pp. 442–448.
13. **Zheng Chen, Shengping Liu, Liu Wenyin, Geguang Pu, and Wei-Ying Ma**, «Building a web thesaurus from web link structure,» in Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, 2003, pp. 48–55.
14. **Z. Harris**, «Distributional Structure,» Word, vol. 10(23), pp. 146–162, 1954.
15. **Hinrich Schutze**, «Word Space,» in Advances in Neural Information Processing Systems , 1992, pp. 895–902.
16. **Carolyn J. Crouch and Bokyung Yang**, «Experiments in automatic statistical thesaurus construction,» in Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval, 1992, pp. 77–88.

17. **Philippovich Y.N. and A.V.Phokhorov**, Semantics of Information Technologies (Semantica Informatsionnikh Technologii: opiti slovarno-tesaurusnogo opisaniya), ISBN 5-8122-0367-9. Moscow, Russia: MGUP, 2002.
18. **Iwayama Makoto, Tanaka Hozumi Tokunaga Takenobu**, «Automatic thesaurus construction based on grammatical relations,» in Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2, pp. 1308–1313.
19. **Sharon A. Caraballo**, «Automatic construction of a hypernym-labeled noun hierarchy from text,» in Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics, 1999 , pp. 120–126.
20. **Dekang Lin and Patrick Pantel**, «Induction of semantic classes from natural language text,» in Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, 2001, pp. 317–322.
21. **Gregory Grefenstette**, «Automatic Thesaurus Generation from Raw Text using Knowledge-Poor Techniques,» in In Making Sense of Words. Ninth Annual Conference of the UW Centre for the New OED and Text Research , 1993.
22. **George K. Zipf**, The Psychobiology of Language.: Houghton-Mifflin, 1935.
23. **J.-P. Chanod, S. Aït-Mokhtar**, «Robustness beyond shallowness: incremental deep parsing,» Natural Language Engineering , vol. 8, no. 3, pp. 121–144, 2002.
24. **Kris Heylen and Dirk Speelman Yves Peirsman**, «Putting things in order: First and second order context models for the calculation of semantic similarity.,» in Proceedings of JADT, Lyon, France, 2008.
25. **Donald Hindle**, «Noun classification from predicate-argument structures ,» in Proceedings of the 28th annual meeting on Association for Computational Linguistics, 1990 , pp. 268–275.
26. **Lynette Hirschman, Ralph Grishman, and and Naomi Sager**, «Grammatically-based automatic word class formation,» Information Processing & Management, vol. 11, no. 1–2, pp. 39–57 , 1975.
27. **Vasileios Hatzivassiloglou and Kathleen R. McKeown**, «owards the automatic identification of adjectival scales: clustering adjectives according to meaning,» in Proceedings of the 31st annual meeting on Association for Computational Linguistics, 1993 , pp. 172–182.
28. **Van der Plas and L.G. Bouma**, «Syntactic contexts for finding semantically related words,» in CLIN, 2004.
29. **Tokunaga Takenobu, Iwayama Makoto, and Tanaka Hozumi**, «Automatic thesaurus construction based on grammatical relations,» in Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2, pp. 1308–1313.
30. **Robert W. Floyd**, «Shortest paths,» Communications of ACM, vol. 5, no. 6, p. 345, 1962.