

Semantic Compression for Text Document Processing

Dariusz Ceglarek^(✉)

Poznan School of Banking, Poznan, Poland
dariusz.ceglarek@wsb.poznan.pl

Abstract. Ongoing research on novel methods and tools that can be applied in Natural Language Processing tasks has resulted in the design of a semantic compression mechanism. Semantic compression is a technique that allows for correct generalization of terms in some given context. Thanks to this generalization a common thought can be detected. The rules governing the generalization process are based on a data structure which is referred to as a domain frequency dictionary. Having established the domain for a given text fragment the disambiguation of possibly many hypernyms becomes a feasible task. Semantic compression, thus an informed generalization, is possible through the use of semantic networks as a knowledge representation structure. In the given overview, it is worth noting that the semantic compression allows for a number of improvements in comparison to already established Natural Language Processing techniques. These improvements, along with a detailed discussion of the various elements of algorithms and data structures that are necessary to make semantic compression a viable solution, are the core of this work. Semantic compression can be applied in a variety of scenarios, e.g. in detection of plagiarism. With increasing effort being spent on developing semantic compression, new domains of application have been discovered. What is more, semantic compression itself has evolved and has been refined by the introduction of new solutions that boost the level of disambiguation efficiency. Thanks to the remodeling of already existing data sources to suit algorithms enabling semantic compression, it has become possible to use semantic compression as a base for automata that, thanks to the exploration of hypernym-hyponym and synonym relations, new concepts that may be included in the knowledge representation structures can now be discovered.

Keywords: Semantic compression · Text clustering · Semantic network · Natural language processing · Information retrieval · Plagiarism detection · Knowledge acquisition

1 Introduction

Intellectual activities related to the gaining of knowledge by people or organizations and the creativity involved in establishing valuable and unique relations are factors that stimulate the formation of intellectual capital. As part of

knowledge management it has been proposed that knowledge resources be publicly available. However, among an organizations information resources there is also sensitive information as well as information that must be protected from falling into the wrong hands. Therefore, knowledge resources are subjected to the rigors of the appropriate security policy, which should prevent information leaks or other forms of infringement of an intellectual property rights. Due to widespread electronic storage, processing and transfer of information, IT systems which are specially designed for this purpose are being more frequently used to protect information resources.

The quality of the functioning of existing mechanisms for protecting such resources depends on many factors. However, it is noticeable that there is a growing tendency for using artificial intelligence algorithms or computational linguistics methods as part of such mechanisms, and that these mechanisms employ increasingly more complex knowledge representation structures (e.g. semantic networks or domain-specific ontologies). The occurrence of intellectual property infringement is constantly on the rise, with corporate and academic resources being particularly under threat. A significant part of corporate information resources is non-confidential and expressed in the form of text and, hence, it cannot be secured by using fingerprinting technologies, as information content itself has intellectual value. In a world that is entirely based on knowledge, intellectual property is contained in various scientific, expert and legal papers. The growing number of incidents connected with the unlawful use of intellectual property is related to increasing access to information. Factors that reinforce the occurrence of such incidents are: constantly easier access to information technologies, free access to information resources stored in an electronic form, especially on the Internet, as well as mechanisms enabling rapid search for information. This is a global phenomenon and it also applies to the academic community with regard to the most sensitive issue appropriation of scientific achievements in all kinds of scientific publications. The existing solutions and systems protecting intellectual property are usually limited to searching for borrowings or plagiarisms in specified (suspicious) documents in relation to documents stored in internal repositories of text documents and, thus, their effectiveness is greatly limited by the size of their repositories.

Detection of borrowings comprises local similarity analysis [25,26], text similarity analysis and global similarity analysis [24,27]. Global similarity analysis uses methods based on citation similarity as well as stylometry, that is, a method of analyzing works of art in order to establish the statistical characteristics of an authors style which helps to determine a works authorship. Text similarity analysis is carried out by checking documents for verbatim text overlaps or by establishing similarity through measuring the co-occurrence of sets of identical words/concepts (based on similarity measures characteristic of retrieval systems). Systems used to detect borrowings differ in terms of the range of sources they search through. These might be public Internet resources or corporate or internal databases; such systems may even use the retrieval systems database indexes. An important attribute of the existing systems is their ability to

properly determine borrowings/plagiarisms in relation to the real number and rate of borrowings. A high level of system precision here means a small number of false positives, whereas a high level of system recall means that it detects the majority of borrowings/plagiarisms in an analyzed set of documents. The most popular existing systems use knowledge representation structures and methods that are characteristic of information retrieval systems processing information for classification purposes at the level of words or strings, without extracting concepts. Moreover, these systems use, as similarity indicators, criteria such as the fact that a coefficient of similarity between documents understood as a set of words appearing in compared documents exceeds several tens of percent and/or the system has detected long identical text passages in a given work. A negative answer from systems of this kind indicating the lack of borrowings in a given document from documents stored in the above-mentioned repositories does not mean that it does not contain borrowings from, for example, documents located at a different Internet address.

The SeiPro2S (Semantically Enhanced Intellectual Property Protection System - SEIPro2S) [22] system was created by the author meets the above-mentioned requirements. Obtained system has been designed to protect resources from the unauthorized use of intellectual property, including its appropriations. The most commonly known example of such treatment is plagiarism, which, as is well known, infringes intellectual property and is based on misappropriation of another person's work or a portion thereof (other people's creative elements) by hiding the origins of that work. In the SeiPro2S system the author used a semantic network as a knowledge representation structure because of its ability to accumulate all the knowledge about the semantics of concepts, which makes it usable in systems that process natural language. It was inspired by works such as [3, 13, 17]. The most important principle of this system is that it will carry out a variety of tasks aimed at protecting intellectual property. The resulting system is not closed and uses both a local repository and all publicly available Internet resources. This mechanism is shown in Fig. 3, whereas the systems architecture and functioning is described in detail in [22]. This system uses a semantic network and follows the so-called text-refinement procedure which is characteristic of the processing of text documents. As part of this procedure it also uses mechanisms such as: text segmentation, morphological analysis, eliminating words that do not carry information, identifying multi-word concepts, disambiguating polysemic concepts. The SeiPro2S system basic task is to determine whether a given text document contains borrowings from other documents (both those stored in a local text document repository and on the Internet).

Furthermore, a suspicious document which is analyzed for borrowings might have undergone various stylistic transformations, which cause it to be regarded by such systems as largely or completely different from other documents as far as long and seemingly unique sentences are concerned. This stylistic transformations include such operation as shuffling, removing, inserting, or replacing words or short phrases or even semantic word variation created by replacing words by one of its synonyms, hyponyms, hypernyms, or even antonyms.

For the above-mentioned reasons there is a need to construct such a mechanism protecting intellectual property contained in text information documents which are expressed in a different way but which have the same information value in semantic terms to be understood as identical or at least very similar.

Semantic compression was designed to handle certain situations, e.g. plagiarism detection in large corpora of documents. Soon afterwards the potential was noticed to enhance the idea of semantic compression and to apply the tools crafted for it to other tasks.

Thus, it became apparent that semantic compression can be a valuable tool in tasks that are in the domain of Information Technology per se, where the chief objective of any technology that can be defined as such, is to present a user with a number of results that fit his or her personal search requirements.

Such a possibility is not straightforward when it comes to pure semantic compression, which can be summarized as **an effective technique for an informed generalization of terms in a given context under additional requirement of minimizing information loss**.

The above summary underlines the need for the recognition of a correct context of any given term. This is a difficult task that probably cannot be completed in 100 % correctly without a sentient mind that would serve as a discriminator equipped with knowledge on not only many of the possible term denotations, but also on the cultural connotations that would allow to draw decisive conclusions.

Nevertheless, the author demonstrated that semantic compression achieves good results when a number of prerequisites is met. The mechanism of semantic compression was implemented in the Semantically Enhanced Intellectual Property Protection System - SEIPro2S [22]. The most important features of semantic compression mechanisms are:

- semantic compression was defined and presented as a technology that can find its place in Natural Language Processing tasks
- the definition and implementation of frequency dictionaries that make semantic compression possible in presence of ambiguity (defined as in [12] along with algorithms supporting the proper choice of a hypernym in a given context [20] (more information available in [5]))
- lossless refactoring of WordNet [15] into WiSENet so that experiments on the quality of semantic compression are possible both in Polish and English documents (introduced in [4])
- highly specialized Finite State Automaton that allows for the automation of building rules allowing for the extraction of data that was not previously defined in the semantic network [11]
- a collaborative approach using local semantic compression which is a specialization of its general case by taking into account frequencies of concepts not in the global case but in the domain that a given document represents (introduced in [6]).

As this work aims to summarize already invested effort and to introduce a number of previously, unpublished results, its structure has to reflect the already available artifacts. Thus, the introductory section is followed by a section

concerning semantic compression and its domain-based variety. The following is a section covering the details of a semantic network that is a base structure for operations of semantic compression itself and its additional applications. Following this is a section devoted to the application of the semantic compression to a semi-automated augmentation of itself through the selection of plausible terms fitting pre-defined patterns. Later, further scenarios in which semantic compression can be used are described along with its advantages as compared to other solutions. The article is concluded with a summary section discussing the obtained results and their quality.

2 Semantic Compression

As was explicitly rendered in the introduction, semantic compression is a technique that has to provide a more general term for the technique in question, i.e. where a term exists in some context that decides on its meaning as seen by a sentient mind. Therefore, when an algorithm has to compute a generalization of a given term it has to carefully choose the degree of generalization. If the generalization is too broad there is considerable information loss. There are cases when this can be a positive feature (as in clustering tasks [16]), but when semantic compression is used to prepare a document for a human user (in a human-readable form) this is not acceptable at all.

Historically, semantic compression was conceived in its global form and later, due to the change in its application domain, it had to be adjusted by further refinement of the generalization strategies. This section presents both varieties along with data obtained from the experiments, thus demonstrating their efficiency and effectiveness.

2.1 Global Semantic Compression

The idea of global semantic compression was introduced by the author in 2010 [5] as a method of improving text document matching techniques both in terms of effectiveness and efficiency. Compression of text is achieved by employing a semantic network and data on term frequencies (in the form of a frequency dictionary). The least frequent terms are treated as unnecessary and are replaced by more general terms (their hypernyms are stored in a semantic network). As a result, the reduced number of terms can be used to represent a text document without significant information loss, which is important from the perspective of processing resources (especially in tasks that require the use of a vector space model [1, 8]).

Furthermore, a reduction in the number of terms helps in dealing with linguistic phenomena, which are problematic in Natural Language Processing [19]. The most commonly referenced phenomena of this type are polysemy and synonymy [12]. When multiple terms used to describe the same or very similar concept occur relatively rarely, they can be replaced by one common, more general, concept. Due to the employment of statistical analysis in the domain

context, already mentioned frequency dictionaries are prepared and let the system deal with polysemic concepts with less effort and a lower error rate than solutions which do not employing such a technique.

As was stated earlier, the procedure of replacing more specific concepts with more general concepts cannot cause significant loss of information. To exemplify this, let us consider a document that is devoted to some biological study. A number of species is mentioned in Latin. Automatic categorization will cause the Latin concepts to extend the vector describing the document, thus they will complicate the whole process. The logical conclusion is that every Latin name of some particular fish can be replaced by a fish concept. As a result, the whole task is carried out with fewer resources and with no significant information loss. Of course, this can only be applied to a specific corpus of documents where these Latin names are rare, thus omissible. The choice of concepts for the generalization is domain-dependent. The data steering this process are in the above mentioned domain frequency dictionaries.

In general, semantic compression enables Natural Language Processing tasks, such as text matching, to operate on a concept level, rather than on a level of individual terms. This can be achieved not only by gathering terms around their common meanings (known from the synset based approach [15]), but also by replacing longer phrases with their more compact forms.

The emphasized concept level allows to capture a **common meaning expressed with a different set of terms**.

Let us demonstrate the idea by introducing some sentences that shall be processed by semantic compression, so that they can be marked as vehicles for the same message. Please note that this is not the result of an actual implementation as it is heavily dependent on the structure and size of the employed semantic network. Actual examples will be given later.

Sentence A. The life span of a cell depends upon the wear and tear on that cell.

Sentence B. Cell's lifetime reposes on accumulated damage.

Sentence A generalized. The period of time of a cell relies on damage on that cell.

Sentence B generalized. Cell's period of time relies on accumulated damage.

The generalizations demonstrated here artificially prepared, yet close to the outputs provided by the SHAPD2 algorithm [23] that was designed by the author. Apart from the fact that the generalization of concepts is not capable of analyzing concept interdependencies and is not able to exchange some phrases according to grammatical rules, they still allow us to make an informed guess that these two differently worded sentences convey some common meaning.

It is worth recalling that semantic compression is a lossy type of compression; yet the loss of information is minimal if the least frequent concepts are selected and are replaced by more general concepts, so their meaning remains as similar to the original meaning as possible. The compression ratio can be tuned easily by setting the number of concepts to be used to describe the text documents. Experiments that were conducted to measure the quality of the method

in Natural Language Processing tasks showed, that the number of words can be reduced to about 4,000 without a significant deterioration of the classification results. The idea of a frequency-driven concept choice is shown in Fig. 1. The goal here is to demonstrate how less general concepts (terms captured in a semantic network) can be replaced by more general concepts. The actual algorithm is given in the following subsection.

2.2 Algorithm for Semantic Compression

Let us assume that, initially, we are given a list k_i of M key concepts, used to create M -dimensional vectors representing the documents, and a target condition: the desired number of key concepts is N (where $N < M$).

First of all, the total frequency of each concept $f(k_i)$ has to be computed for all documents in the documents corpus. This is achieved by computing the cumulated term frequency, i.e. by adding the sum of hyponyms frequencies to the frequency of the hypernym. In the second step, incorporation of information from the synonymy relation is carried out in which the synonym with the largest cumulated frequency is chosen. Finally, the terms with the largest cumulative frequency are selected. Moving upwards in the hierarchy, the cumulative concept frequency is calculated by adding the sum of hyponyms' frequencies to the frequency of the hypernym: $cumf(k_i) = f(k_i) + \sum_j cumf(k_j)$, where k_i is a hypernym of k_j - in pseudocode Algorithm 1. The cumulated frequencies are to be sorted and the N concepts with top values are selected as target key concepts (descriptor list) - see Algorithms 2 and 3.

Finally, the algorithm defines the *compression mapping rules* for the remaining $(M - N)$ words in order to handle every occurrence of k_j as its hypernym k_i in further processing. If necessary (when a hypernym has not been selected as a descriptor), the mapping rules can be nested.

This is essential as it allows to shorten individual vectors by replacing terms with lower information capacity by their descriptors (refer to Fig. 1). The described method of semantic compression results in a reduction of vector space dimensions by $(M - N)$. As an result, a part of the specific information, which is proportional to the information capacity of concepts not selected as descriptors, is lost.

2.3 Global Semantic Compression Evaluation

It is now time to carefully describe a situation in which semantic compression can be applied. Let us imagine that a document is an artifact to be matched against a corpus of other documents. This can take place in a variety of occasions; one of them is the intellectual property system (such as SEIPro2S). In order to apply semantic compression it is postulated that the system is equipped in various domain-specific corpora. These corpora let the system come up with a set of word frequencies that is specific to some area of interest (medicine, computer science, mathematics, astronomy, biology, etc.). To illustrate this, let us consider the following scenario. When the system processes a document that is a piece of news concerning recent advances in antibiotics research which has been

Algorithm 1. Selection of a concept used to represent those generalized concepts, followed by calculation of the cumulated concept instance frequency in document corpus C

```

 $S(v)$  - set of synonyms for a concept  $v$ 
 $V$  - vector of concepts stored in the semantic network
 $V'$  - topologically sorted vector  $V$ 
 $V''$  - reversed vector  $V'$ 
 $l_v$  - number of occurrences of a concept  $v$  in corpus  $C$ 
 $H_v$  - set of hypernyms for a concept  $v$ 
//choosing representing synonym
 $max = 0$ 
 $n = 0$ 
 $sum = 0$ 
for  $s \in S(v)$  do
     $sum = sum + l_s$ 
    if  $l_s > max$  then
         $max = l_s$ 
         $n = s$ 
    end if
end for
 $l_n = sum$ 
//calculating cumulated frequency for hypernymy
for  $v \in V''$  do
     $p = card(H_v)$ 
    for  $h \in H_v$  do
         $l_h = l_h + \frac{l_v}{p}$ 
    end for
end for

```

posted in some popular magazine, we can take advantage of the domain corpora. Establishing a document domain is a simple task, that takes into account the variety of tools provided by the NLP; yet it will allow us to undertake additional steps.

When both the potential reader and the system are aware of the document type we can use the semantic network to compress concepts. When we come back to the scenario with the news concerning advances in antibiotics we can safely assume that this is not a highly specialized article. Thus, any reference to penicillin or ampicillin is a possible point where semantic compression can be applied. The semantic network stores data on concepts and their mutual relationships - a good semantic network will store data that will reflect the fact that penicillin is a type of antibiotic, as is ampicillin.

The result of applying semantic compression is visible in shortening the global vector by 3 elements (see Fig. 1). Instead of entries for an antibiotic, penicillin and ampicillin, we can store just the first entry. Analogical actions should be performed for concepts that are too specific in the context of the processed document.

Algorithm 2. Choosing N concepts in a domain-compressed semantic network

```

 $L$  - vector storing number of concept occurrences in document corpus  $C$ 
 $L'$  - vector  $L$  sorted in a descending order
 $f$  - number of occurrences of  $m$ -th concept in vector  $L'$ 
for  $v \in L$  do
  if  $l_v \geq f$  then
     $d_v = v$ 
  else
     $d_v = FMax(v)$ 
  end if
end for

```

Algorithm 3. FMax procedure - finding a descriptor for a hypernym with the highest frequency

```

FMax( $v$ ):

 $max = 0$ 
 $x = \emptyset$ 
for  $h \in H_v$  do
  if  $d_h \neq \emptyset$  then
    if  $l_{dh} > max$  then
       $max = l_{dh}$ 
       $x = d_h$ 
    end if
  end if
end for
return  $x$ 

```

The author devised an experiment to verify whether semantic compression does indeed yield better results when applied to specific text-processing tasks there was devised an experiment. The evaluation experiment was performed by making a comparison of the clustering results for texts that were not semantically compressed with those that were [10]. For the experiment was used a document corpus consisting of documents coming from the following domains: *business, crime, culture, health, politics, sport, biology and astronomy*.

In order to verify the results, all of the documents were initially labeled manually with a category. All of the documents were written in English.

The clustering procedure was performed 8 times. The first run was done without the semantic compression mechanism: all of the identified concepts (about 25000 - this is only about a fifth of all the concepts in the research material) were included. Then the semantic compression algorithm was used to gradually reduce the number of concepts; it started with 12000 and preceded with 10000, 8000, 6000, 4000, 2000 and 1000 concepts.

The classification results were evaluated by being compared with the labels specified by the document editors: the ratio of correct classifications was calculated [1,9]. The outcome is presented in Tables 1 and 4. The loss of classification

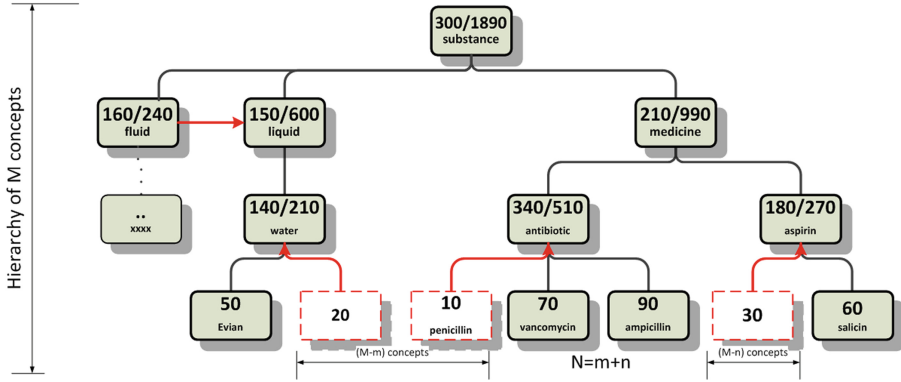


Fig. 1. Selection of N concepts with top cumulative frequencies

Table 1. Classification quality for two runs, the upper line denotes the results when semantic compression was enabled

Clustering features	1000	900	800	700	600	Average
All concepts	94.78 %	92.50 %	93.22 %	91.78 %	91.44 %	92.11 %
12000 concepts	93.39 %	93.00 %	92.22 %	92.44 %	91.28 %	91.81 %
10000 concepts	93.78 %	93.50 %	93.17 %	92.56 %	91.28 %	92.23 %
8000 concepts	94.06 %	94.61 %	94.11 %	93.50 %	92.72 %	93.26 %
6000 concepts	95.39 %	94.67 %	94.17 %	94.28 %	93.67 %	93.95 %
4000 concepts	95.28 %	94.72 %	95.11 %	94.56 %	94.06 %	94.29 %
2000 concepts	95.56 %	95.11 %	94.61 %	93.89 %	93.06 %	93.96 %
1000 concepts	95.44 %	94.67 %	93.67 %	94.28 %	92.89 %	93.68 %

quality is virtually insignificant for a semantic compression strength which reduces the number of concepts to 4000.

As was briefly remarked in an earlier section the conducted experiment indicates that the semantic compression algorithm can be employed in classification tasks to significantly reduce the number of concepts and the corresponding vector dimensions. As a result, tasks with extensive computational complexity are performed more quickly.

A set of examples of semantically compressed text fragments (for 4000 chosen concepts) is now given. Each compressed fragment is presented after the original fragment.

- 1a** The information from AgCam will provide useful data to agricultural producers in North Dakota and neighboring states, benefiting farmers and ranchers and providing ways for them to protect the environment
- 1b** information will provide adjective data adjective producer American state adjective state benefit creator provide structure protect environment.

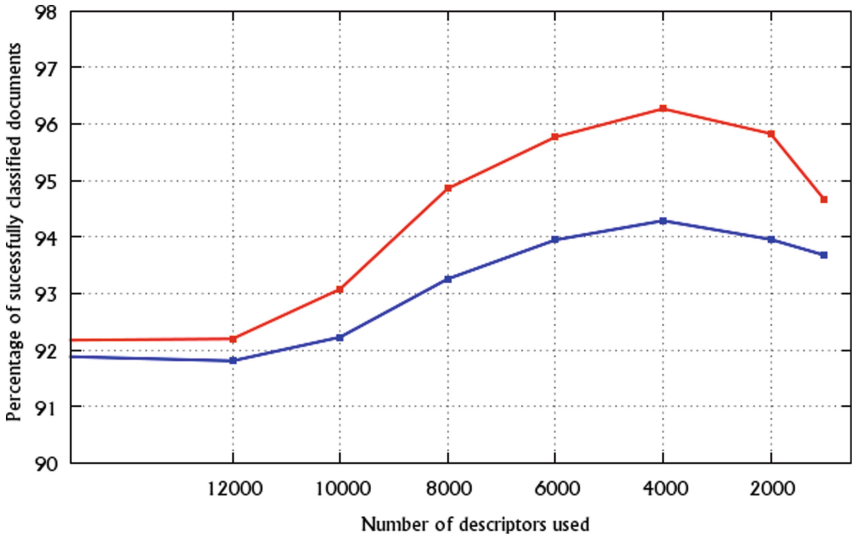


Fig. 2. Classification quality for two runs, upper line denotes results when the semantic compression was enabled

- 2a** Researchers trying to restore vision damaged by disease have found promise in a tiny implant that sows seeds of new cells in the eye. The diseases macular degeneration and retinitis pigmentosa lay waste to photoreceptors, the cells in the retina that turn light into electrical signals carried to the brain
- 2b** researcher adjective restore vision damaged by-bid disease have found predict tiny implant even-toed ungulate seed new cell eye disease macular degeneration retinitis pigmentosa destroy photoreceptor cell retina change state light electrical signal carry brain.
- 3a** Together the two groups make up nearly 70 % of all flowering plants and are part of a larger clade known as Pentapetalae, which means five petals. Understanding how these plants are related is a large undertaking that could help ecologists better understand which species are more vulnerable to environmental factors such as climate change
- 3b** together two group constitute percent group flowering plant part flowering plant known means five leafage understanding plant related large undertaking can help biologist better understand species more adjective environmental factor such climate change (Fig. 2).

Figure 3 presents the average evaluation results from two classification tasks. The loss of classification quality is virtually insignificant for a semantic compression strength which reduces the number of concepts to 6000. Stronger semantic compression and further reduction of the concept number entails a deterioration of the classification quality (which can, however, be still acceptable).

The conducted experiment indicates that the semantic compression algorithm can be employed in classification tasks in order to significantly reduce the

Table 2. Evaluation of a classification with semantic compression task 1 (780 documents) results.

Clustering features	1000	900	800	700	600	Average
Without SC	93.46 %	90.90 %	91.92 %	92.69 %	89.49 %	91.69 %
12000 concepts	91.92 %	90.38 %	90.77 %	88.59 %	87.95 %	89.92 %
10000 concepts	93.08 %	89.62 %	91.67 %	90.51 %	90.90 %	91.15 %
8000 concepts	92.05 %	92.69 %	90.51 %	91.03 %	89.23 %	91.10 %
6000 concepts	91.79 %	90.77 %	90.90 %	89.74 %	91.03 %	90.85 %
4000 concepts	88.33 %	89.62 %	87.69 %	86.79 %	86.92 %	87.87 %
2000 concepts	86.54 %	87.18 %	85.77 %	85.13 %	84.74 %	85.87 %
1000 concepts	83.85 %	84.10 %	81.92 %	81.28 %	80.51 %	82.33 %

Table 3. Evaluation of a classification with semantic compression task 2 (900 documents) results.

Clustering features	1000	900	800	700	600	Average
Without SC	93.78 %	93.89 %	93.11 %	92.56 %	92.11 %	92.03 %
12000 concepts	93.00 %	94.00 %	94.00 %	91.33 %	90.78 %	91.49 %
10000 concepts	93.33 %	94.22 %	93.56 %	93.44 %	92.22 %	92.33 %
8000 concepts	92.78 %	93.22 %	94.22 %	93.33 %	90.89 %	91.79 %
6000 concepts	92.56 %	93.44 %	92.22 %	92.89 %	91.00 %	91.26 %
4000 concepts	92.00 %	92.44 %	91.22 %	90.89 %	90.22 %	90.03 %
2000 concepts	92.33 %	91.78 %	89.89 %	90.56 %	89.67 %	89.44 %
1000 concepts	92.00 %	92.00 %	88.33 %	87.11 %	83.78 %	86.90 %

number of concepts and the corresponding vector dimensions. As a result, tasks with extensive computational complexity are performed faster (with linearithmic complexity).

To summarize, semantic compression is more effective when a text domain is identified and an appropriate domain frequency dictionary is used to perform the process. It should be emphasized that the more compact the context frame is, the better. Ideally, the context frame that decides about which frequency dictionary should be used, should be coverage on a one-to-one basis with a single sentence. Unfortunately, due to the previous observations, this is not possible.

2.4 Domain Based Semantic Compression

Global semantic compression combines data from two sources: the term frequencies from the frequency dictionary and the concept hierarchy from the semantic network. Usually one extensive semantic network is used for a given language

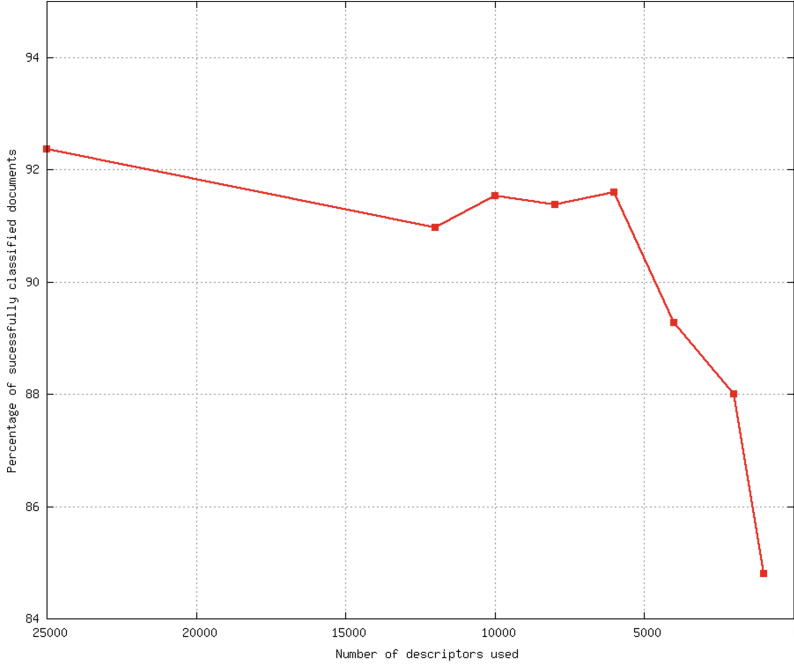


Fig. 3. Experiments' results of classification quality

(e.g. WiSENet [7] for English based on WordNet [15], SenecaNet for Polish [22]) and thus it is able to include linguistic knowledge covering multiple domains.

The varying characteristics based on the term frequency in some domain have a crucial impact on functioning of the semantic compression's algorithm, as term frequencies are a decisive factor in identifying the least frequent terms to be replaced by their hypernyms in a given domain.

In order to give an illustrative example we will consider a document originating from nature studies where the life of common rodents is discussed. On the other hand, let us consider a document from Information Technology which is focused on Human Computer Interfaces. Both documents have passages where the concept *mouse* is to be found. Domain-based semantic compression allow to generalize the concept *mouse* into two different less specific concepts, i.e. different concepts for every domain. In nature studies the concept *mouse* can be generalized as a rodent while when dealing with the term in Information Technology it can be regarded as an electronic device. Global semantic compression with a low number of final output concepts, thus a high level of compression, would choose the hypernym according to the overall frequency which might introduce unnecessary chaos.

Further experiments conducted by the author confirmed, that natural language tasks employing domain-based semantic compression yield better results

Table 4. Classification quality using semantic compression with an enabled proper names dictionary enabled

Clustering features	1000	900	800	700	600	Average
All concepts	94.78 %	92.50 %	93.22 %	91.78 %	91.44 %	92.11 %
12000 concepts	93.56 %	93.39 %	93.89 %	91.50 %	91.78 %	92.20 %
10000 concepts	95.72 %	94.78 %	93.89 %	91.61 %	92.17 %	93.08 %
8000 concepts	95.89 %	95.83 %	94.61 %	95.28 %	94.72 %	94.86 %
6000 concepts	96.94 %	96.11 %	96.28 %	96.17 %	95.06 %	95.77 %
4000 concepts	96.83 %	96.33 %	96.89 %	96.06 %	96.72 %	96.27 %
2000 concepts	97.06 %	96.28 %	95.83 %	96.11 %	95.56 %	95.83 %
1000 concepts	96.22 %	95.56 %	94.78 %	94.89 %	94.00 %	94.66 %

when its general form is used. As was exemplified, this is because domain frequency dictionaries better reflect language characteristics.

Domain-based semantic compression is easily implementable with already crafted artifacts as long as a certain procedure is applied to the processed text. When the domain is established for a text fragment, a specific domain frequency dictionary can be applied to perform local generalization by avoiding extreme cases and touching concepts that are too general in the context of a given domain. This differentiates domain-based semantic compression from global compression, as the latter maximizes savings in terms of a possibly shortest length of a vector that represents the processed documents. Yet, in the envisioned test application, avoiding the introduction of unfamiliar (from the domain point of view) concepts is an important advantage that is readily exploited for the benefit of the interested users.

Thus, the proposed procedure was tested on a group of users in order to verify applicability of Domain-Based Semantic Compression. This was done to measure whether it increased the level of text comprehension. This measurement cannot be done without a human user, due to the elusive nature of evaluating the explored problem. As an outcome the author gathered feedback from users participating in the experiment.

To summarize the results, it should be emphasized that the applying a locally adjusted domain frequency dictionary improves readability and allows for exclusion of generalized concepts that do not fit into the context. Additional comments offered by the surveyed users contain the following remarks:

- domain-based semantic compression can better fit the text context and allow for less misunderstandings
- it uses concepts that are less ambiguous, thus it allows for better understanding of the text
- in many cases global semantic compression causes a generalized concept to have a different meaning

- O godzinie 19:42:06 Księżyc dotknie cienia Ziemi. Stopniowo od wschodniej strony nasz satelita będzie "pożerany" przez cień naszej planety. O godzinie 20:49:34 cień całkowicie pochłonie Księżyc. Jego barwa powinna stać się krwisto czerwona na skutek oświetlenia promieniami słonecznymi zagiętymi w ziemskiej atmosferze. Maksimum zaćmienia wypadnie o godzinie 21:20:36.
- O godzinie 19:42:06 Księżyc dotknie cienia Ziemi. Stopniowo od wschodniej strony nasz satelita będzie **konsumowany** przez cień naszej planety. O godzinie 20:49:34 cień całkowicie **przyłączy** Księżyc. Jego barwa powinna stać się **kolorowo** czerwona na skutek **działania** promieniami słonecznymi **nierównymi** w ziemskiej atmosferze. Maksimum zaćmienia **usunie** o godzinie 21:20:36.

Fig. 4. Sample question from the user survey: the Polish version was used throughout the experiments

- global semantic compression produces a text that is perceived as being unrelated and whose meaning is unclear
- global semantic compression introduces concepts there are outside of the domain.

The performed experiment consisted of a set of four samples presented to the participants. There were 32 participants in the experiment. The participants that were surveyed had not received any training in the domains that were in the samples presented to them. Every sample comprised 3 text fragments. The first fragment was an unmodified text fragment taken from a corpus at random. The only constraint that every fragment had to follow was that its domain should be as unambiguous as possible. The chosen fragment was then transformed first by domain-based semantic compression and second, it was transformed by global semantic compression. The participant had to make a choice of whether he or she preferred the first transformation or the second transformation. He or she had to make a choice three more times and at the end share his or her opinions on his or her decisions and the motivation behind them. The sample for Polish is given in Fig. 4 and for English in Fig. 5. Please note that the experiment was in Polish and these were solutions that were discussed in their initial form.

The whole experiment was conducted using the SenecaNet semantic network and extensions derived from the project Morfologik [14] project. Inclusion of the Morfologik dictionary allowed for automatic changes of declination and conjugation of the word forms. This is a difficult task for languages such as Polish due to the large number of forms that are possible for every word (including gender, person and grammatical aspect). Effort was made to achieve over 95.5 % of correct transformations where every transformation was a two-phase process. It is worth noting that for an error the author understands an error to be a change from verb or adjective to noun. First, a concept was identified, then it was checked whether it could be a candidate for generalization. Secondly, when a

There is new promise on the horizon for those who suffer from REM Sleep Behaviour Disorder (RBD) according to researchers at the University of Toronto. RBD, a neurological disorder that causes violent twitches and muscle contractions during rapid eye-movement (REM) sleep, can lead to serious injuries. John Peever, Assistant Professor at the University of Toronto, discovered that an inhibitory brain chemical called glycine is responsible for actively suppressing muscle twitches in REM sleep. Deficiency in glycine levels in the brain cells that control muscles (motoneurons) was found to cause the violent muscle contractions that mimic the primary symptom of RBD. This study shows the mechanism that suppresses muscles twitches in REM sleep and this will lead to better treatments and potential cures for this disorder, says Peever. Treating REM sleep disorder may have much broader implications, since within five to eight years of being diagnosed with this disorder, 60-80% of individuals eventually develop Parkinsons disease. Source : University of Toronto

There is new promise on the scope for those who suffer from physical condition state demeanor change RBD according to researchers at the University of Toronto. RBD, a nervous disorder that causes violent **symptoms** and step-down during physical condition physical condition can lead to serious injuries. John Peever, professor at the University of Toronto, ascertained that an inhibitory brain chemical called organic compound is responsible for actively suppressing go across **symptoms** in physical condition lack in organic compound levels in the nerve cell that control go across nerve cell was found to cause the violent step-down that mimic the chief symptom of RBD. This study shows the mechanism that suppresses go across **symptoms** in physical condition and this will lead to better treatments and potential **medicaments** for this change says Peever. Treating physis(twitches) ay have much adult female reasoning since within five to eight years of being analyze with this change 60-80% go on to manifest many develop Parkinsons disease. Source : University

Fig. 5. Sample text from applying domain-based semantic compression on an English corpus

concept was generalized, a proper form had to be applied in order to present user performing an evaluation with a maximally uncluttered text fragment. Excerpts from the survey and the transformed text are given for Polish and English.

3 Semantic Network as a Key Data Structure for Semantic Compression

As was earlier emphasized, any reasonable text transformation that promises informed choices when substituting one term for another one of a more general nature that fits into the text’s domain must be based on a structure capable of storing a variety of semantic relations.

A number of structures ranging from simple dictionaries through thesauruses to ontologies were applied in these types of categories [21]. Out these the semantic network was proven to be the best solution due to its outstanding features coupled with a lack of unnecessary complexity.

The WiSENet is a semantic network that captures data from the WordNet but these data are structured in a manner following that of the SenecaNet. The features of the SenecaNet, WiSENet and the transformation procedure along with a detailed discussion on various implementation details are given below.

3.1 SenecaNet Features and Structure

The SenecaNet is a semantic network that stores relations among concepts for Polish. It stores over 156400 concepts, other features are listed in Table 5. This was the first semantic network to be used in Semantic Compression.

It meets the requirements of a semantic network in every aspect. The concepts are represented as a list. There is a specific format that allows for fast traversal and a number of check-up optimizations that the SenecaNet implements. Each

Table 5. Comparison of the WordNet and the SenecaNet semantic networks

Features	WordNet	SenecaNet
Concept count	155200	156400
Polysemic words	27000	22700
Synonyms	0	8330
Homonyms, hypernoms	+	+
Antonyms	−	+
Connotations	+	+
Unnamed relationship	−	+

entry is stored in a way that allows to reference connected concepts in an efficient manner. Every entry from this list conveys information on the actual descriptor to be found in the text as well as the hypernoms, synonyms, antonyms and descriptors that are in an unnamed relation to the given descriptor.

There is an additional rule that every descriptor can occur exactly one time on the leftmost part of the entry when the whole semantic network is considered. This restriction introduces an extremely important feature, i.e. there can be no cycles in a structure devised in this manner.

Each descriptor can have one or more hypernoms (a heterarchy as in [21]). Each descriptor can have one or more synonyms. The synonyms are listed only once on the right side of the entry, they do not occur on the leftmost part of entry, as this is an additional anti-cycle guard.

An excerpt from the WiSENet format is given below in Table 6 to illustrate the described content.

Table 6. Example of the SenecaNet notation

Barack Obama \leftarrow *politician*, #*president(USA)*, #*citizen_of(USA)*, ϵ *Noun*
 car \leftarrow *vehicle*, &*engine*, ϵ *Noun*
 gigabyte \leftarrow *computermemoryunit*, &*byte*, ϵ *Noun*
 Real Madrid \leftarrow *footballteam*, @*Madrid*, ϵ *Noun*
 volume unit \leftarrow *unitofmeasurement*, @*volume*, ϵ *Noun*
 Jerusalem \leftarrow *city*, : *PalestineAuthority*, ϵ *Noun*
 Anoushka Shankar \leftarrow *musician*, #*sitarist*, #*daughter(RaviShankar)*, ϵ *Noun*
 Hillary Clinton \leftarrow *politician*, #*secretaryofstate(USA)*, #*citizen_of(USA)*, ϵ *Noun*

For many applications the structure of semantic network is transparent, i.e. it does not affect the tasks the net is applied to. Nevertheless, semantic compression is much easier when the descriptors are represented by actual terms and when their variants are stored as synonyms.

Algorithm 4. Algorithm for the WordNet to the SenecaNet format (WiSENet) transformation

```

WN - WordNet, as a list of synsets identified by descriptors d
S - synset, containing multiple lemmas l
F[l] - number of synsets containing lemma l
SN - output WiSENet structure
for all (d, S) ∈ WN do
    for all l ∈ S do
        F[l] ++
    end for
end for
for all (d, S) ∈ WN do
    parse lemma from synset descriptor
    l = split(d, "." ) [0]
    if F[l] = 1 then
        lemma can be used as synset descriptor
        d = l
    else
        for all l ∈ S do
            if F[l] = 1 then
                d = l
            exit
            end if
        end for
    end if
    SN[d] = S
end for

```

3.2 WordNet to SenecaNet Conversion

When faced with the implementation of semantic compression for English one has to use a solution that has similar capabilities as those on SenecaNet. Building up a new semantic network for English is a great task that would surpass the author's capabilities, thus he turned to existing solutions. The WordNet proved to be an excellent resource, as it was applied by numerous research teams to a great number of tasks which yielded good results.

The author had to confront the challenge of converting a synset-oriented structure into new semantic network without cycles operating on the descriptors in order to be recognized as actual concepts in the processed text fragments. An algorithm to accomplish this has been devised. It operates on sets by taking into account data on every lemma stored in a given synset and synsets (therefore their lemmas) that are hypernyms to the processed synset.

The synset is understood as a group of concepts that have similar meaning. Under close scrutiny many concepts gathered in one synset fail to be perfect synonyms to each other. They share a common sense, yet the degree to which they do varies. A lemma is any member of the synset; it can be a single concept or a group of concepts representing some phrase [15].

Before the algorithm is given, an example of a naive approach to the problem is demonstrated. This shall enable the reader to follow the process of a semantic network transformation in greater detail and with less effort.

In order to avoid graph cycles in the target structure, the author needed to modify the way one chooses terms to describe a synset. The best situation is when a lemma contained in a synset descriptor belongs only to that synset, i.e. the lemma itself is a unique synset descriptor. In other situations the author tried to find another lemma from the same synset which would satisfies the condition. The experiments proved that this produces the desired networks, but cannot satisfy the criterion for a lack of losses during the transformation. The obtained semantic network consisted of only 25000 terms serving as concepts, where a total of 86000 noun synsets were processed. Eventually, a “synthetic” synset descriptor was developed. The introduction of synthetic descriptors is not contrary to the author’s ambitions to convert the WordNet into WiSENet in a lossless manner along with using actual concepts as concept descriptors. Synthetic descriptors are always the result of untangling of some cycle, thus, they can always be outputted as actual concepts to be found in the processed text.

Please refer to Figs. 6 and 7 to see a visualization of this process. Please notice that the term “approximation” is contained in several synsets: thus, it fails as a concept descriptor (see Fig. 6). One can easily observe that the term “bringing close together” occurs exactly once, thus it can replace the synthetic descriptor “approximation.n.04”.

All of this is gathered in Tables 7 and 8.

Table 7. Companion table for Fig. 6

Synset	Terms	Parent synset
change of integrity	change of integrity	change.n.03
joining.n.01	joining, connection, connection	change of integrity
approximation.n.04	approximation, bringing close together	joining.n.01
approximation.n.03	approximation	version.n.01
approximation.n.02	approximation	similarity.n.01
estimate.n.01	estimate, estimation, approximation, idea	calculation.n.02

In order to remedy the issues as detailed above, the procedure to transform the WordNet structure efficiently is given below. For a pseudocode description please refer to the listing in Algorithm 4.

The first step of the procedure is to build a frequency dictionary (F) for lemmas by counting the synsets containing a given lemma. The algorithm loops through all of the synsets in WordNet (WN), and all the lemmas in the synsets (S) and counts every lemma occurrence. In the second step it picks a descriptor (possibly a lemma) for every synset. Next, it begins checking whether a synset descriptor (d) contains a satisfactory lemma. After splitting the descriptor (the

Table 8. Companion table for Fig. 7

Term	Parents
change of integrity	change.n.03
approximation	bringing close together, approximation.n.02, estimate.n.01, approximation.n.03
approximation.n.02	similarity.n.01
approximation.n.03	version.n.01
bringing close together	joining
joining	change of integrity
estimate.n.01	calculation.n.02
estimate	estimate.n.02, estimate.n.01, estimate.n.05, appraisal.n.02, estimate.n.04, compute, count on

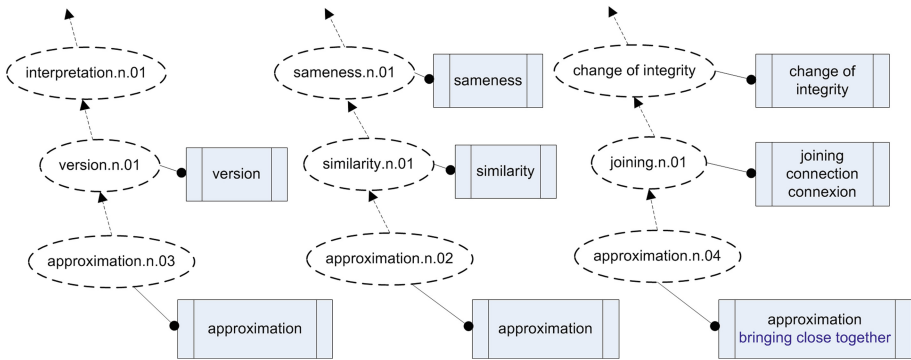


Fig. 6. WordNet synset description

partition point is the first dot in the synset description) and taking the first element of the resulting list the algorithm examines, whether such a lemma occurs exactly once throughout all of the synsets - if the answer is positive then it can be used as a new synset descriptor. If it is not, it loops through the lemmas from the examined synset and checks if there is any unique lemma which can be utilized as a descriptor. In case no unique lemma is found, a genuine WordNet descriptor is used.

4 Applications

One of the most important applications where Semantic Compression can be used is the semi-automated expansion of itself. Another preparation, presented as a viability test of Domain-Based Semantic Compression are generalized documents presented in a human-readable form and in a way that correlates with

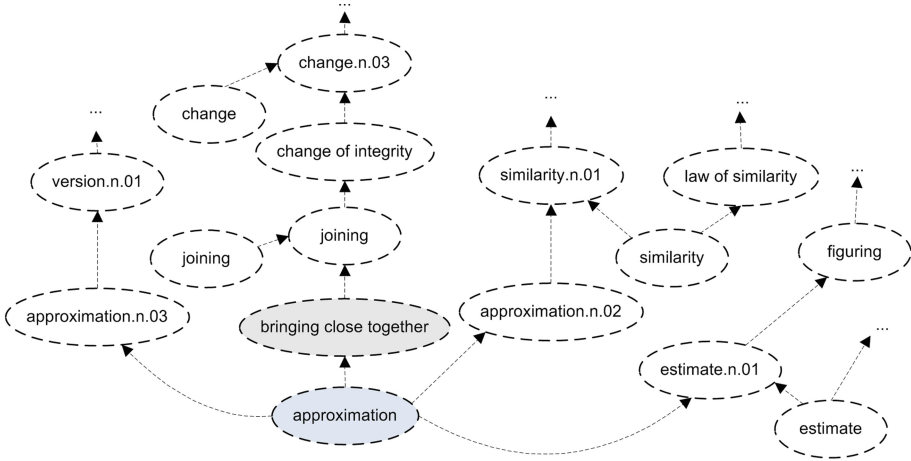


Fig. 7. Concepts description in SenecaNet format

the user's preferences and his/her knowledge of a given domain. There are other applications that will be discussed briefly in the summarizing section of this work.

4.1 Semantic Network Expansion

Exhaustive experiments led to the conclusion that there have to be seed data in order to capture potential unknown concepts. Seed data are understood as a number of concepts fitting into a given rule slot. The key challenge was the bootstrapping nature of all the experiments. One wants a more exhaustive model, and there is no easy way to sidestep the investment of time and effort into this task. Ideally, one could use a fairly large body of skilled specialists that would extend WiSENet by including data stored in various resources. Unfortunately, the author could not afford to do this, thus the methods that have been devised strive for maximum output with minimum input. This can be true even if the minimum input in some cases can be as much as 40000 of specific proper name concepts used as seed data.

4.2 Algorithm for Matching Rules

All of the operations are performed with WiSENet as the structure containing the necessary concepts. The first important step in the algorithm is a procedure that unwinds the rule into all of the hyponyms stored inside the network. This operation can be a considerable cost in terms of execution as it has to traverse all possible routes from a chosen concept to the terminal nodes in the network. After completion a list of rules is obtained, listing every possible permutation of the concepts from the semantic network. To shorten the processing time, one can

specify the number of levels that procedure shall descend in during its course of execution.

The next phase of the algorithm is to step through the textual data in order to find matches on the computed rules. The stepping through is done by employing a bag of concepts algorithm. The bag of concepts was implemented as a Finite State Automaton with advanced methods for triggering desired actions (automaton working as a transducer). The transducer allows the SenecaNet network to become of a comparable size and performance quality as the WordNet which carries out NLP tasks. In addition, through the use of an unnamed relationship in SenecaNet, the quality of the mechanism finding the proper meaning of ambiguous terms increased. The tasks being carried out by the SeiPro2S system for English are made possible through the adoption of WordNet and by having it adjusted to already existing tools crafted for SEIPro2S. At any state it checks whether any of the rules to be matched is completed. The automated method of expansion of new concepts and new lexical relationships in the SenecaNet network uses a specially constructed transducer. A discussion covering the details of transducer implementation is beyond the scope of this article. Nevertheless, it can be visualized as a frame passing through the textual data. With every shift towards the end of text fragment, concepts inside the frame are used to check whether they trigger any of the rules obtained in the first phase. The size of the bag is chosen by the researcher, yet the performed experiments show that the best results are obtained for a bag sized 8–12 when the rules are 2–5 concepts long.

The bag of concepts algorithm is a good idea, as it tolerates mixins and concept order permutations. All matchings are performed after the initial text processing phase has been performed. The text processing phase (also called the *text refinement procedure*) consists of well-known procedures such as applying a stop list and term normalization.

A mixin is in this case a passage of text that serves some purpose to original text, yet it separates two or more concepts that exist in one of the computed rules.

Consider the following examples:

Rule - disease (all hyponyms), therapy (all hyponyms)

Match in: chemotherapy drug finish off remaining cancer

Matched concepts: therapy -> chemotherapy, disease -> cancer

Mixin: drug finish off remaining

Match in: gene therapy development lymphoma say woods

Matched concepts: therapy -> gene therapy, disease -> lymphoma

Mixin: development

Match in: cancer by-bid using surgery chemotherapy

Matched concepts: therapy -> chemotherapy, disease -> cancer

Mixin: by-bid using surgery

The examples are taken from one of the experiments performed with a biology corpus. It can be observed that the bag of concepts performs well in various cases, as it handles long mixins and concept permutation. An additional observation should be made that concepts which were hyponyms to those in the original example rule were matched (as was referenced earlier).

All the experiments that were performed took into account the possibility of matching more than a single rule. Thus, a mechanism for triggering a set of rules was devised and was signaled earlier along with the bag of concepts.

The procedure of matching rules holds internal registers which store rules that are actively valid with a given bag of concepts. To give an example, please consider a set of three rules:

rule 1: university, city (all hyponyms)

rule 2: university, city (all hyponyms), country (all hyponyms)

rule 3: person (all hyponyms), academic.

A given exemplary text fragment: *A team of chemists led by chemistry professor David Giedroc from Indiana University (in Bloomington, USA) described a previously unknown function of a protein they now know is responsible for protecting a major bacterial pathogen from toxic levels of copper. Co-author with Giedroc on the paper is professor Michael J. Maroney of the University of Massachusetts. The results were published Jan. 27 in Nature Chemical Biology.*

The procedure will match and matches previously defined rules:

rule number 1 with university → university, Bloomington → city, *newconcept: Indiana University in Bloomington*

rule number 2 with university → university, Bloomington → city, USA → country, *newconcept: Indiana University in Bloomington*

rule number 3 with David → first name, professor → academic, *newconcept: David Giedroc = professor(Indiana University, University in Bloomington)*

rule number 3 with Michael → first name, professor → academic, *newconcept: Michael J. Maroney = professor(University of Massachusetts).*

When a complete rule or its part is mapped, it is presented to the user to accept the match or reject it. The user can decide whether he or she is interested in total matches all partial matches. When the bag of concepts drops earlier concepts and is filled with new concepts, the rules that were not matched are dropped from the register of valid rules. The whole process of matching rules is presented in Fig. 8.

The algorithm in pseudocode is presented in listing 5.

4.3 Experiment with Semantic Compression Based Pattern Matching

The devised algorithm was used to perform an experiment on biology-related data. The test corpus consisted of 2589 documents. The total number of words in the documents was over 9 million. The essential purpose of the experiment

Algorithm 5. Algorithm for matching rules using WiSENet and bag of concepts

```

SN – Semantic Network
R – semantic relation pattern
BAG – currently active bag of concepts
Rule – set of processed rules

//attach rule triggers to concepts in semantic network
mapRulesToSemNet(SN, R[])
for all Rule ∈ R do
  for all Word, Relations ∈ Rule do
    N = getNeighbourhood(SN, Word, Relations)
    for all Word ∈ N do
      createRuleTrigger(SN, Word, Rule)
    end for
  end for
end for

//text processing: tokenization, phrases, stop list
T = analyzeText(Input)
for all Word ∈ T do
  if count(BAG) = size(BAG) then
    //first, deactivate rules hits for a word
    //that drops out from bag of words
    oldWordpop(Bag)
  end if
  for all Rule ∈ getTriggers(SN, oldWord) do
    unhit(Rule, Word)
    push(Bag, Word)
    for all Rule ∈ getTriggers(SN, Word) do
      //take all relevant rules and activate word hit
      hit(Rule, Word)
      if hitCount(Rule) = hitRequired(Rule) then
        //report bag of words when hits reaches required number
        report(Rule, Bag)
      end if
    end for
  end for
end for

```

was to find specialists and their affiliations. This converges with the motivating scenario, as WiSENet was enriched by specialists (and their fields of interest), universities, institutes and research centers. The experiment used the following rules:

rule 1 first name (all hyponyms), professor (all hyponyms), university (all hyponyms)

rule 2 first name (all hyponyms), professor (all hyponyms), institute (all hyponyms)

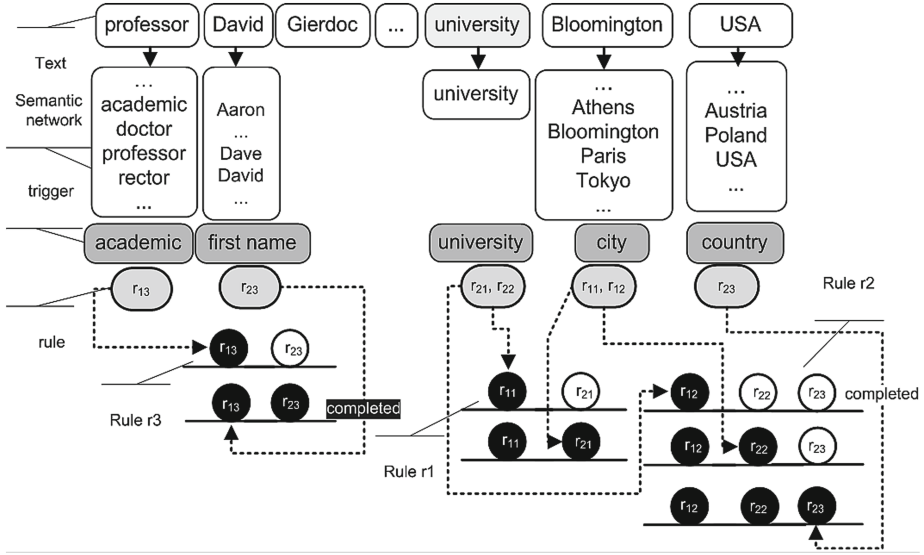


Fig. 8. Process of matching rules from the example.

- rule 3** first name (all hyponyms), professor (all hyponyms), research center (all hyponyms)
rule 4 first name (all hyponyms), professor (all hyponyms), department (all hyponyms)
rule 5 first name (all hyponyms), professor (all hyponyms), college (all hyponyms).

The size of the bag of concepts was set at 8 elements. Additionally, all rules were to match exactly all of the concepts.

Out of 1326 documents where a concept “professor” was found, the prepared rules matched 445 text fragments. This gives a recall rate of 33.56 %. Precision of results was 84.56 %. This level was found to be very satisfactory, especially when taking into account that, due to the algorithm, there can be duplicates of matched text fragments (due to the multiple triggering of rules inside the current bag of concepts).

In addition, the experiment resulted in 471 concepts that were previously unknown to WiSENet. The context and type of rules that matched the text fragments led to extremely efficient updates of the semantic network.

Table 9 demonstrates the sample results from the experiment. Please note that a match on its own does not discover new concepts. The rules present potential fragments that with high likelihood, contain new concepts that can be included into the semantic network.

Table 9. Sample results of experiments with rules based on the WiSENet on a corpus of biology-related documents. The discovered concepts are written under the matches. Multiple activated rules were omitted.

Text fragment	Match/discovered concept	Rule
Explain senior author Douglas Smith Md professor department neurosurgery director	Douglas professor department Douglas Smith	5
Feb proceedings national academy of sciences researcher University of Illinois entomology professor Charles Whitfield postdoctoral	University of Illinois professor Charles Charles Whitfield	1
Design function biological network she-bop visiting professor Harvard University Robert Dicke fellow visiting	Professor Harvard University Robert Robert Dicke	1
Modify bacteria Thomas Wood professor –Artie– –McFerrin– department chemical engineering have	Thomas professor department Thomas Wood	5
Matthew –Meyerson– professor pathology Dana –Farber– cancer institute senior associate	Matthew professor institute Matthew Meyerson	2
An assistant professor medical oncology Dana –Farber– cancer institute researcher broad assistant	Professor Dana institute Dana Farber	2
Vacuole David Russell professor molecular microbiology –Cornell’s– college veterinary medicine colleague	David professor college David Russell	4

5 Conclusions

Research efforts on developing and extending semantic compression and its applications can be considered as valuable. What is more, the article presents a variety of new research artifacts such as:

- formulation of the notion of semantic compression
- rules for the preparation of frequency dictionaries
- the bag of concepts algorithm
- the system for domain-based semantic compression
- transforming the English semantic network WordNet into a SenecaNet format (WiSENet)
- the algorithm for pattern matching with semantic compression
- the automaton of pattern matching.

All of the above achievements were used in a number of experiments that tested various characteristics. It was proven that the clustering of documents that underwent semantic compression was more efficient than the same procedure on a corpus of uncompressed data. The increase on efficiency for already good results (the average was 92.11 %) amounted to an additional 4.16 %).

What is more, domain-based semantic compression tested as a live system with active participants demonstrated that semantic compression aided by resources such as Morfologik proved to satisfactory with its results to users.

Semantic compression-based patterns are an interesting option for the retrieval of concepts that were previously unknown to a semantic network. What is more, the syntax of the patterns is straightforward, and possibly anyone understanding the idea of a less or more general concept can use it to design, own patterns that can be fed into the pattern matching system.

References

1. Baeza-Yates, R.A., Ribeiro-Neto, B.: Modern Information Retrieval. Addison-Wesley Longman Publishing Co. Inc., Boston (1999)
2. Boyd-Graber, J., Blei, D.M., Zhu, X.: A topic model for word sense disambiguation. In: EMNLP (2007)
3. Burrows, S., Tahaghoghi, S.M.M., Zobel, J.: Efficient plagiarism detection for large code repositories. *Softw.: Pract. Exper.* **37**(2), 151–175 (2007)
4. Ceglarek, D., Haniewicz, K., Rutkowski, W.: Quality of semantic compression in classification. In: Pan, J.-S., Chen, S.-M., Nguyen, N.T. (eds.) ICCCI 2010, Part I. LNCS, vol. 6421, pp. 162–171. Springer, Heidelberg (2010)
5. Ceglarek, D., Haniewicz, K., Rutkowski, W.: Semantic compression for specialised information retrieval systems. In: Nguyen, N.T., Katarzyniak, R., Chen, S.-M. (eds.) Advances in Intelligent Information and Database Systems. SCI, vol. 283, pp. 111–121. Springer, Heidelberg (2010)
6. Ceglarek, D., Haniewicz, K., Rutkowski, W.: Domain based semantic compression for automatic text comprehension augmentation and recommendation. In: Jędrzejowicz, P., Nguyen, N.T., Hoang, K. (eds.) ICCCI 2011, Part II. LNCS, vol. 6923, pp. 40–49. Springer, Heidelberg (2011)
7. Ceglarek, D., Haniewicz, K., Rutkowski, W.: Towards knowledge acquisition with WiSENet. In: Nguyen, N.T., Trawiński, B., Jung, J.J. (eds.) New Challenges for Intelligent Information and Database Systems. SCI, vol. 351, pp. 75–84. Springer, Heidelberg (2011)
8. Erk, K., Padó, S.: A structured vector space model for word meaning in context. In: EMNLP, pp. 897–906. ACL (2008)

9. Frakes, W.B., Baeza-Yates, R.A. (eds.): *Information Retrieval: Data Structures & Algorithms*. Prentice-Hall, Upper Saddle River (1992)
10. Hotho, A., Staab, S., Stumme, G.: Explaining text clustering results using semantic structures. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) *PKDD 2003. LNCS (LNAI)*, vol. 2838, pp. 217–228. Springer, Heidelberg (2003)
11. Khan, L., McLeod, D., Hovy, E.: Retrieval effectiveness of an ontology-based model for information selection. *VLDB J.* **13**, 71–85 (2004)
12. Krovetz, R., Croft, W.B.: Lexical ambiguity and information retrieval. *ACM Trans. Inf. Syst.* **10**, 115–141 (1992)
13. Lukashenko, R., Graudina, V., Grundspenkis, J.: Computer-based plagiarism detection methods and tools: an overview. In: *Proceedings of the 2007 International Conference on Computer Systems and Technologies, CompSysTech '07*, New York, NY, USA, pp. 40:1–40:6. ACM (2007)
14. Mikowski, M.: Automated building of error corpora of polish. In: Lewandowska-Tomaszczyk, B. (ed.) *Corpus Linguistics, Computer Tools, and Applications State of the Art, PALC 2007*, pp. 631–639. Peter Lang, Frankfurt am Main, Berlin, Bern, Bruxelles, New York, Oxford, Wien, (2008)
15. Miller, G.A.: WordNet: a lexical database for english. *Commun. ACM* **38**, 39–41 (1995)
16. Nock, R., Nielsen, F.: On weighting clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(8), 1223–1235 (2006)
17. Ota, T., Masuyama, S.: Automatic plagiarism detection among term papers. In: *Proceedings of the 3rd International Universal Communication Symposium, IUCS '09*, pp. 395–399, New York, NY, USA. ACM (2009)
18. Sanderson, M.: Word sense disambiguation and information retrieval. In: Croft, W.B., van Rijsbergen, C.J. (eds.) *SIGIR '94*, pp. 142–151. ACM/Springer, London (1994)
19. Sinha, R., Mihalcea, R.: Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In: *ICSC*, pp. 363–369. IEEE Computer Society (2007)
20. Snow, R., Jurafsky, D., Ng, A.Y.: Learning syntactic patterns for automatic hypernym discovery. In: *Advances in Neural Information Processing Systems (NIPS 2004)*, November 2004. This is a draft version from the NIPS preproceedings; the final version will be published by April 2005
21. Staab, S., Hotho, A.: Ontology-based text document clustering. In: Klopotek, M.A., Wierzchon, S.T., Trojanowski, K. (eds.) *Intelligent Information Processing and Web Mining. Advances in Soft Computing*, vol. 22, pp. 451–452. Springer, Heidelberg (2003)
22. Ceglarek, D.: Architecture of the semantically enhanced intellectual property protection system. In: Burduk, R., Jackowski, K., Kurzynski, M., Wozniak, M., Zolnierrek, A. (eds.) *CORES 2013. AISC*, vol. 226, pp. 711–720. Springer, Heidelberg (2013)
23. Ceglarek, D.: Single-pass corpus to corpus comparison by sentence hashing. In: Badica, A., Trawinski, B., Nguyen, N.T. (eds.) *Recent Developments in Computational Collective Intelligence - Concepts. Applications and Systems*, volume 7092 of *Studies in Computational Intelligence*, pp. 167–177. Springer, Heidelberg (2013)
24. Hoad, T.C., Zobel, J.: Methods for identifying versioned and plagiarized documents. *J. Am. Soc. Inf. Sci. Technol.* **54**(3), 203–215 (2003)
25. Charikar, M.S.: Similarity estimation techniques from rounding algorithms. In: *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing, STOC '02*, pp. 380–388. ACM (2002)

26. Manber, U.: Finding similar files in a large file system. In: Proceedings of the USENIX Winter 1994 Technical Conference on USENIX Winter 1994 Technical Conference, WTEC'94, Berkeley, CA, USA, p. 2. USENIX Association (1994)
27. Stein, B., Lipka, N., Prettenhoferr, P.: Intrinsic plagiarism analysis. *Lang. Resour. Eval.* **45**(1), 63–82 (2010). Springer, Netherlands

Transactions on Computational Collective Intelligence
XIV

Nguyen, N.T. (Ed.)

2014, IX, 197 p. 71 illus., Softcover

ISBN: 978-3-662-44508-2