Automatic Extraction of Semantic Similarity of Words from Raw Technical Texts

Aristomenis Thanopoulos, Nikos Fakotakis, George Kokkinakis

Wire Communications Laboratory
Electrical & Computer Engineering Dept., University of Patras
265 00 Rion, Patras, Greece
{aristom,fakotaki,gkokkin}@wcl.ee.upatras.gr

Abstract

In this paper we address the problem of extracting semantic similarity relations between lexical entities based on context similarities as they appear in specialized text corpora. Only general-purpose linguistic tools are utilized in order to achieve portability across domains and languages. Lexical context is extended beyond immediate adjacency but is still confined by clause boundaries. Morfological and collocational information are employed in order to exploit the most of the contextual data. The extracted semantic similarity relations are transformed to semantic clusters which is a primal form of a domain-specific term thesaurus.

1. Introduction

The unceasing expansion of human activity to new thematic areas in science, society and culture requires a corresponding expansion of the necessary linguistic resources so that, both humans and machines, are able to process effectively relevant information. In the context of lexical knowledge acquisition, lexical semantics extraction is not only the most demanding task but also crucial for many applications, such as Language Modeling, Information Retrieval, Information Extraction, Machine Translation, Word Sense Disambiguation, Thesauri construction, etc. The abundance of electronic text in several natural languages and many thematic domains allows us to employ automatic corpus processing methods in order to extract lexical semantics. The development of automatic approaches is important since they offer rapid acquisition or modification of lexical resources in new or dynamically evolved sublanguages. Moreover, word meanings are domain-dependent in general and humans do not always succeed in classifying special meanings that everyday words appear in specific thematic contexts.

The main approaches proposed for the extraction of lexical level semantic knowledge are *syntax*-based and *n-gram*-based. Syntax-based methods (Pereira & Thishby, 1992; Grefenstette, 1993; Li & Abe, 1997) represent the words as vectors containing statistic values of their syntactic properties in relation to a given set of words (e.g. statistics of *object* syntax relations referring to a set of verbs) and cluster the derived vectors according to the commonality of these properties. Methods that use bigrams (Brown et al., 1992) or trigrams (Martin et al., 1998) cluster words considering as a word's context the one or two immediately adjacent words and employ as clustering criteria the minimal loss of average mutual information and the perplexity improvement respectively. Such methods are oriented to language modeling and aim primarily at rough clustering of large vocabularies.

2. Problem Setting

In this paper we address the problem of automatic extraction of semantic similarity relations between lexical items in relational form from which fine-grained hierarchical clusters are obtained. In order to restrict the vocabulary and the word ambiguity and to utilize information-rich technical texts, processing is confined to corpora from specific domains. This restriction is acceptable in the framework of NLP systems, which are usually operating on sub-languages and interested only in domain-specific word meanings. Human-readable thesauri (should) provide as well semantic relations of words in relevance to thematic domains. Therefore, we aim at developing a method applicable to every domain for which specific corpora are available in order to extract domain-dependent word meaning relations.

In order to achieve portability we chose to approach the problem from a knowledge-poor perspective. N-gram methods, which share the same perspective, focus on fast processing of large corpora and consider as context only the immediately adjacent words without exploiting medium-distance word dependencies. Since large corpora are available only for few domains we aimed at developing a method for processing small or medium sized corpora exploiting as much as possible contextual information rich in semantic restrictions. Our approach was driven by the observation that in constrained domain corpora the vocabulary and the syntactic structures are limited and that small or medium distance word or phrase patterns are often used to express similar facts.

Stock market financial news and Modern Greek, were used as domain and language test case respectively. Throughout the paper examples taken from English corpora are used as well.

3. Extraction of Semantic Classes

A fundamental issue on context-based distributional word clustering methods is the definition of the effective scope of context. Although most approaches use local context of one or two words we employ a more flexible notion of context: We consider that syntactic relations which impose semantic constraints rarely exceed clause boundaries. Even when they do, advanced linguistic techniques, as ellipsis and anaphora resolution, are required to detect them. Therefore we utilize contextual data extended up to clause boundaries.

We detect clause boundaries as: a) Sentence boundaries using a sentence segmentation tool (Stamatatos et al., 1999). b) Intrasentential clause boundaries based on detection of specific conjunctions which anambiguously introduce them.

The main idea supporting context-based word clustering is that two words that can substitute one another in several different contexts always providing meaningful

word sequences are probably semantically similar. Present n-gram based methods utilize this assumption considering as word context only the one or two immediately adjacent words.

In the present work we generalize the concept of words to that of semantic tokens. Therefore we consider as context of a semantic token the surrounding semantic tokens belonging to the same clause.

Roughly sketched, the employed algorithm for constructing semantic classes is as follows:

1. Tokenize "semantically" the corpus classifying lexical items or chains to initial semantic classes.
2. Gather context statistics about the derived semantic tokens in terms of their context tokens.
3. Estimate semantic similarity between tokens based on their context similarity.
4. Cluster together semantic tokens that are confidently estimated as semantically similar.
5. Repeat steps 2-4 using the extracted semantic classes until a terminating criterion is met.

3.1. Extracting Initial Semantic Classes

Since technical corpora are usually limited in size special treatment is required in order to overcome the problem of sparse contextual data. A first step to this direction is to initially classify lexical items or patterns belonging to known and domain-independent semantic classes. Therefore the number of parameters is considerably reduced while contextual data are increased accordingly. Context-free tools are applied to classify word tokens or token sequences to common semantic categories. This preprocessing task reffered to from now on as "semantic tokenization" includes the following procedures:

1. A lemmatizer and part-of-speech tagger is applied in order to assign to every word its corresponding lemma and syntactic category, since all words produced from the same lemma are classified to the same semantic category and detection of semantic relations is confined among words of the same syntactic category.
2. Pattern matching: Known and domain-independent patterns (e.g. dates, numbers, amounts, etc.) are regarded as single semantic tokens and classified to respective categories. Their specific information content is of no interest to our task; therefore only the corresponding class is maintained.
3. Frequently appearing noun phrase patterns that represent single semantic entities are treated as a single token (e.g. "Dow Jones", "minister of industry", "*NUMBER* years old", "χρήση *YEAR*" (= annual results of YEAR), etc.). Their detection is based on the relatively high value of the mutual information of adjacent semanticful lexical items (excluding verbs and adverbs) as calculated by:

$$I_{mutual}(w_1, w_2) = \log \frac{P(w_1 w_2)}{P(w_1)P(w_2)}$$

4. Chaining together extracted bigrams that constitute word chains frequently encountered in the corpus a set of rigid noun phrases is created. From these the spurious ones are manually discarded to ensure that the detected n-grams actually represent semantic entities.

Henceforth by "***semantic token***" we refer to the recognized semantic category of a pattern (e.g. <*date*> for the string *"3/5/1999"*), the recognized lemma of a content word (e.g. *"increase"* for *"increased"*), a lemmatized rigid word chain (e.g. *"mutual_fund"* for *"mutual funds"*) or an unrecognized word. By "***lexical items***" we refer to the latter three.

3.2. Context Similarity Estimation

Counting the number of occurrences of every semantic token found in the corpus we define a frequency threshold under which no semantic clustering is attempted. So only Frequent Semantic Entities (FSE) are subjected to clustering (except the SEs represented in the corpus by known patterns), while all but the most rare Semantic Tokens are used as clustering parameters. The corresponding frequency thresholds in the presented experiments were set to 20 and 10 respectively in order to acquire sufficient contextual data for every FSE constraining computational time. Ideally, any word appearing at least twice in the corpus should be used as context parameter.

Definite determiners and verb auxiliaries are excluded from the processing since they have no semantic connection with their head words while pronouns are handled as semantically empty words.

In order to extract context similarity estimation about FSEs (algorithm steps 1-2) we employ two different algorithms:

3.2.1. Context-Vector Similarity Estimation Algorithm

As Semantic Word-based Context (SWC) of a semantic token in a given text we define any of the adjacent semantic tokens (without exceeding clause boundaries), each one specified by its signed distance (in number of semantic tokens) from the considered token.

1. For every FSE in the corpus, SWC statistics are gathered (i.e. the number of times a specific token was encountered in a specific signed distance from the considered token).

2. A weighted Tanimoto measure (Charniak, 1993), is employed and tested for the context similarity estimation between FSEs, calculating the similarity between 2 items as the ratio of the number of their parameters in common (in our case the total number of their common SWCs) divided by the number of their total parameters (their total context STs) and multiplied with a weight function which depends on the proximity of the corresponding context token. We define this function as inversely proportional to the "contextual distance" between the FSE in question and the corresponding adjacent semantic token (i.e. the context parameter). However this measure was found having the drawback that handles all contextual data in a uniform way extracting spurious results in cases where few similar contexts appear many times, usually due to often-used stereotyped expressions or repeated facts. Instead preference should be given to hits derived from many different pairs of similar contexts. This was obtained applying a logarithmic function to the numerator of the above ratio. Finally, summing over all context parameters, we obtain a measure expressing the context similarity between two semantic tokens.

3.2.2. Pattern-Matching Algorithm

In the previously described method the notion of word context is based on independent intra-clausal contextual data. In the course of research it became apparent that similar contextual patterns are a much more reliable similarity criterion than single token occurrences. That is, if the clausal contexts of 2 tokens have at least two elements in common, then we count this as a hit regarding the similarity of the 2 tokens. Since the patterns under detection vary across languages and domains we need a method that extracts them dynamically, independently of the text genre.

In order to avoid the storage of intermediate information (i.e. context statistics) described in CVSE algorithm, we employ the cross-correlation algorithm in order to directly extract lexical similarity hypotheses from text.

We borrowed the cross-correlation concept from the signal processing domain where it is used to detect similarities between 2 signals. In the text processing domain, a clause can be considered as a digital signal in which every semantic token corresponds to a signal sample. In order to detect words with common contexts each clause is checked on matching each other partially. This check is performed on every possible relative position between them. If common patterns of semantic tokens are found, the tokens on the two clauses which are similarly adjacent to the patterns are indicated as candidate semantic relatives. In order to determine the semantic cross-correlation between 2 clauses, common patterns of semantic tokens are detected.

Consider, for example, the following sentences:

In the third quarter, managed portfolios typically showed no growth while ***the S&P index inched up 0.3****.*

In Hong Kong, the Hang Seng Index inched up 180.60 *to finish at 2601.70.*

Since the two clauses in boldface have been detected and tokenized as:

[<*S&P*> <*index*> <*inched_up*> <NUMBER>]

[<in> <hong_kong> <*hang_seng*> <*index*> <*inched_up*> <NUMBER>]

their common pattern

[<*index*> <*inched_up*> <NUMBER>]

indicates a probable semantic connection between the tokens <*S&P*> and <*hang_seng*>. The pairwise context similarity function between them is augmented by an additive term, calculated from:

$$F(s_i, s_j) = \sum_{p}^{x} \frac{1}{|d_p|} + \sum_{\substack{q,p \\ q<p}}^{x} \frac{1}{\sqrt{|d_p \cdot d_q|}}$$

where d_p, (p=1..X) the distance of the token s_i (or equivalently, s_j) from every (1 to x) constituent of the common pattern.

Keeping only the first term we obtain the same result as in the CVSE method with weight function h(d)=1/|d|. The second term augments the score in proportion to the cohesion, the size and the relative position (to the semantic token in question) of the common pattern.

During this process contextual data are not maintained in memory; instead the detection of a common pattern in both sentences results to the storage of several hits in general (i.e. candidate similar pairs of tokens) or to the increase of their corresponding similarity measure according to the pattern similarity of their contexts.

In order to reduce the computational time and required memory during the whole process a pruning mechanism is applied at regular time intervals to eliminate word pairs with a relatively very low semantic similarity score.

Finally, keeping the N-best scoring pairs, we obtain the preponderant semantically related candidates.

3.3. Hierarchical Clustering

Although the obtained set of similarity relations between semantic tokens constitutes already a (thesaurus-like) form of semantic similarity representation, most NLP applications require semantic clusters instead of semantic similarity relations. Since a semantic similarity measure has been extracted, the formation of semantic classes is an ordinary clustering problem.

In order to construct semantic classes we apply an unsupervised agglomerative hard clustering algorithm to process the set of the extracted semantic relations:

1. Each semantic token is initially assigned to a cluster.
2. Clusters are merged together according to their distance until a final condition is met. Tracking the successive word-to-class assignments we obtain sub-cluster hierarchies as shown in Figure 1. The calculation of semantic distance between two clusters is based on the *average linkage measure*, that is the average distance between the items in one cluster and the items in another cluster. No merging of clusters is realized if the distance between the closest clusters is less than a threshold proportional to the average distance between all objects under clustering.

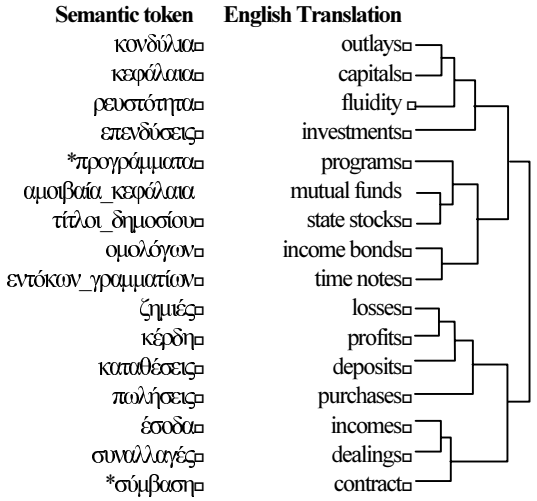

Figure 1: A derived sample hierarchial cluster of lexical entities[1].

3.4. Iterative Processing

The described similarity estimation algorithms, both PM and (V)CVSE, produce a set of semantic similarity relations between lexical items that represent semantic entities. However contextual data are not sufficient enough for all lexical entities to yield reliable results.

[1] The words appear in their most frequent in the corpus morphological form and their translation is domain-specific.

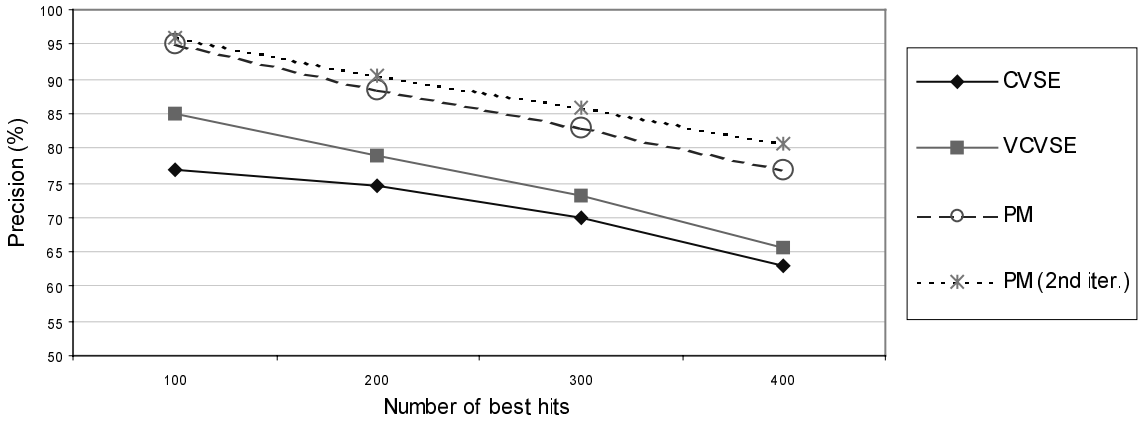

Figure 2. Comparative results of the employed semantic similarity extraction algorithms

Sample Clusters after 1st iteration	Sample Clusters after 2nd iteration
υποχώρησε (fell), ενισχύθηκε (augmented)	υποχώρησε, ενισχύθηκε, τερμάτισε, κυμαίνεται
frf, chf, itl, dem, gbp, jpy	frf, chf, itl, dem, gbp, jpy, usd, Νικκει, DAX, CAC40, δολάριο, FTSE, dow_jones
Νικκει, DAX, CAC40, δολάριο (dollar), FTSE	άνοδο, απώλειες, ανοδικές, πτωτικές, αύξηση, πτώση
τερμάτισε (closed at), κυμαίνεται (fluctuates)	κοινές (common), προνομιούχες (preference shares), ανώνυμες, ονομαστικές
άνοδο (rise), απώλειες (losses)	εταιρία (company), όμιλο (group), διοίκηση (management), μονάδα (unit), μέτοχοι (shareholders)
ανοδικές (upward), πτωτικές (downward)	

Table 1: Sample clusters derived after 1st and 2nd iteration.

Lexical entities occurring less frequently or appearing various lexical functions in the corpus do not always obtain proper semantic connections. In order to utilize more contextual data for such words we initially apply the similarity estimation algorithm and make use only of the top scoring relations. The clustering algorithm operates on a portion of the semantic entities and produces clusters with very high precision. The previously described procedures (similarity estimation & hierarchical clustering) are applied iteratively using the tags of the previously extracted semantic clusters as semantic tokens. Having reduced the number of parameters, more dense contextual data are exploited and more accurate semantic relations and clusters are extracted. Since initial clustering errors are propagated to the next iterations degrading severely the final outcome, special care must be taken in order to avoid them, either by choosing a rather high cut-off similarity score or by manually discarding erroneous semantic relations.

4. Experimental Results

The reported experiments have been carried out on a 220.000 words corpus, comprised of financial news of 1998 which was constructed in the framework of a currently carried out R&D project for Information Extraction from raw text[2]. The evaluation of the results was performed by checking manually the validity of extracted similarity relations and clusters. The semantic similarity estimation algorithms (described in section 3.2) were tested and their precision was measured. The results indicate that context similarity detection based on pattern-matching yields more reliable results than the vector similarity method. This demonstrates the importance of the cross-correlation procedure, which is the only computationally feasible method for pattern similarity detection.

One iterative loop was performed using the PM algorithm. Sample clusters after first and second iteration are shown on Table 1.

The precision of all tested algorithms in detecting semantic similarity relations in connection with the number of the best scored relations maintained can be seen on Figure 2.

Regarding the clustering procedure, a set of about 200 FSEs, from the 700 subjected to clustering, was partitioned to 40 clusters, each one hierarchically structured. Considering a typical cluster formed (Figure 1) we note that from the 16 semantic tokens that constitute the cluster all but two (denoted with '*') refer to forms of money investment or profit. From the vocabulary of semantic tokens subject to clustering 4 tokens belonging to the same class were not detected; therefore accuracy and recall for the specific cluster are 87.5% and 78,8% respectively.

5. Discussion

In the most similar approach to the one presented here Redington et al. (1993) used a rigid local context of ±2 words. The CVSE algorithm we developed is more advanced in that a more flexible concept of context is employed. However the pattern matching algorithm obtains significantly better results. We conclude that the PM Algorithm outperforms (V)CVSE in that employing a

[2] Project "MITOS" of the Greek General Secretariat for Research and Technology

pattern-based notion of context instead of simple context adjacency statistics, information-richer structures of natural language are exploited, without the need of specialized linguistic tools that syntax-based approaches require. Moreover, new corpora can be used directly to imporve the outcome without processing anew the updated statistical data.

However the VCVSE Algorithm offers the possibility to express the semantic similarity estimation as a function of individual context parameters and therefore to base similarity estimation to variety of context similarity. CVSE Algorithm requires extraction of contextual data only once from the corpus. Iterative applications require only the update of the context vectors simply by merging parameters.

6. Conclusions

Initiating from the concept of word semantic similarity estimation in terms of context similarity we have proposed an approach for extracting semantic similarity relations between lexical entities and then semantic clusters by processing word adjacency data obtained from small or medium sized corpora.

The innovations of the present work are the dynamic restriction of context parameters inside clause boundaries, the usage of rigid-phrase and word-morphology based initial semantic categories and the pattern-matching cross-correlation algorithm which dynamically detects pattern context similarity offering strong evidence for semantic similarity.

Although the presented method features language and domain portability we consider that the genre of the texts consisting the training corpora should be constraint to news or reports, preferably with small variation regarding their source, in order to achieve reliable results.

In the immediate future we plan to apply a soft clustering algorithm to the extracted relations, to test the method to a different domain and/or language and to proceed to quantitative comparison with other word clustering methods.

References

Brown P.F., DellaPietra V.J., DeSouza P.V., Lai J.C., and Mercer R.L., 1992. Class-Based n-gram Models of Natural Language. *Computational Linguistics*, 18(4):467-479.

Charniak E., 1993. *Statistical Language Learning*, The MIT Press.

Grefenstette, G., 1993. SEXTANT: Extracting Semantics fromRaw Text: Implementation Details. *The Journal of Knowledge Engineering.*

Li H., Abe N., 1997. Clustering Words with the MDL Principle. *Journal of Natural Language Processing* 4(2).

Martin S., Liermann J., Ney H., 1998. Algorithms for bigram and trigram word clustering. *Speech Communication,* 24:19-37.

McMahon J.G., Smith F.J., 1996. Improving Statistical Language Model Performance with Automatically Generated Word Hierarchies. *Computational Linguistics*, 22(2).

Pereira F., Tishby N., 1992. Distributional Similarity, Phrase Transitions and Hierarchical Clustering. *Working Notes, Fall Symposium Series*. AAAI 1:108-112.

Redington, F. M., Chater, N., and Finch, S., 1993. Distributional Information and the Acquisition of Linguistic Categories: A Statistical Approach. *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*, 1:848-853.

Stamatatos E., Fakotakis N., Kokkinakis G., 1999. Automatic Extraction of Rules for Sentence Boundary Disambiguation. *Proceecings of ACAI*, Workshop on Machine Learning in Human Language Technology, 1:88-92.