7

Learning to Build a Semantic Thesaurus from Free Text Corpora without External Help

Katia Lida Kermanidis
Ionian University
Greece

1. Introduction

The automatic extraction and representation of domain knowledge has been attracting the interest of researchers significantly during the last years. The plethora of available information, the need for intelligent information retrieval, as well as the rise of the semantic web, have motivated information scientists to develop numerous approaches to building thesauri, like dictionaries and Ontologies that are specific to a given domain.

Ontologies are hierarchical structures of domain concepts that are enriched with semantic relations linking the concepts together, as well as concept properties. Domain terms populate the ontology, as they are assigned to belong to one or more concepts, and enable the communication and information exchange between domain experts. Furthermore, domain Ontologies enable information retrieval, data mining, intelligent search, automatic translation, question answering within the domain.

Building Ontologies automatically to the largest extent possible, i.e. keeping manual intervention to a minimum, has first the advantage of an easily updateable extracted ontology, and second of largely avoiding the subjective, i.e. biased, impact of domain experts, which is inevitable in manually-based approaches.

This chapter describes the knowledge-poor process of extracting ontological information in the economic domain mostly automatically from Modern Greek text using statistical filters and machine learning techniques. Fig. 1 shows the various stages of the process. In a first stage, the text corpora are being pre-processed. Pre-processing includes tokenization, basic morphological tagging and recognition of named and other semantic entities, that are

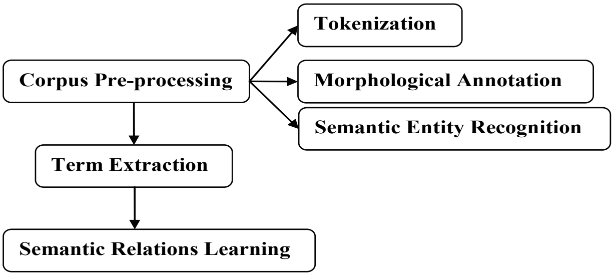

Fig. 1. System overview

Source: Machine Learning, Book edited by: Abdelhamid Mellouk and Abdennacer Chebira,
ISBN 978-3-902613-56-1, pp. 450, February 2009, I-Tech, Vienna, Austria

related to the economic domain (e.g. values, amounts, percentages etc), and that would be useful in future data-mining applications. In a second stage, content-words in the text are categorized into domain terms and non-terms, i.e. words that are economic terms and words that aren't. Finally, domain terms are linked together with various types of semantic relations, such as hyponymy/hyperonymy (is-a), meronymy (part-of), and other relations of economic nature that don't fit the typical profile of *is-a* or *part-of* relations.

2. Comparison to related work

As mentioned earlier, significant research effort has been put into the automatic extraction of domain-specific knowledge. This section describes the most characteristic approaches for every stage in the process, and compares the proposed process to them.

Regarding named entity recognition, Hendrickx and Van den Bosch (2003) employ manually tagged and chunked English and German datasets, and use memory-based learning to learn new named entities that belong to four categories. They perform iterative deepening to optimize their algorithmic parameter and feature selection, and extend the learning strategy by adding seed list (gazetteer) information, by performing stacking and by making use of unannotated data. They report an average f-score on all four categories of 78.20% on the English test set. Another approach that makes use of external gazetteers is described in (Ciaramita & Altun, 2005), where a Hidden Markov Model and Semi-Markov Model is applied to the CoNLL 2003 dataset. The authors report a mean f-score of 90%. Multiple stacking is also employed in (Tsukamoto et al., 2002) on Spanish and Dutch data and the authors report 71.49% and 60.93% mean f-score respectively. The work in (Sporleder et al., 2006) focuses on the Natural History domain. They employ a Dutch zoological database to learn three different named-entity classes, and use the contents of specific fields of the database to bootstrap the named entity tagger. In order to learn new entities they, too, train a memory-based learner. Their reported average f-measure reaches 68.65% for all three entity classes. Other approaches (Radu et al., 2003; Wu et al., 2006) utilize combinations of classifiers in order to tag new named entities by ensemble learning.

For the automatic extraction of domain terms, various approaches have been proposed in the literature. Regarding the linguistic pre-processing of the text corpora, approaches vary from simple tokenization and part-of-speech tagging (Drouin, 2004; Frantzi et al., 2000), to the use of shallow parsers and higher-level linguistic processors (Hulth, 2003; Navigli & Velardi, 2004). The latter aim at identifying syntactic patterns, like noun phrases, and their structure (e.g. head-modifier), in order to rule out tokens that are grammatically impossible to constitute terms (e.g. adverbs, verbs, pronouns, articles, etc). The statistical filters, that have been employed in previous work to filter out non-terms, also vary. Using corpus comparison, the techniques try to identify words/phrases that present a different statistical behaviour in the corpus of the target domain, compared to their behaviour in the rest of the corpora. Such words/phrases are considered to be terms of the domain in question. In the simplest case, the observed frequencies of the candidate terms are compared (Drouin, 2004). Kilgarriff (2001) experiments with various other metrics, like the χ^2 score, the t-test, mutual information, the Mann-Whitney rank test, the Log Likelihood, Fisher's exact test and the TF.IDF (term frequency-inverse document frequency). Frantzi et al. (2000) present a metric that combines statistical (frequencies of compound terms and their nested sub-terms) and linguistic (context words are assigned a weight of importance) information.

In the field of taxonomy learning, previous approaches have varied from supervised to unsupervised clustering techniques, and from methodologies that make use of external taxonomic thesauri, to those that rely on no external resources. Regarding previous approaches that employ clustering techniques, Cimiano et al. (2004) describe a conceptual clustering method that is based on the Formal Concept Analysis for automatic taxonomy construction from text and compares it to similarity-based clustering (agglomerative and Bi-Section-KMeans clustering). The automatically generated ontology is compared against a hand-crafted gold standard ontology for the tourism domain and report a maximum lexical recall of 44.6%. Other clustering approaches are described in (Faure & Nedellec, 1998) and (Pereira et al., 1993). The former uses a syntactically parsed text (verb-subcategorization examples) and utilize iterative clustering to form new concept graphs. The latter also makes use of verb-object dependencies, and relative frequencies and relative entropy as similarity metrics for clustering. Pekar and Staab (2002) take advantage of a taxonomic thesaurus (a tourism-domain ontology) to improve the accuracy of classifying new words into its classes. Their classification algorithm is an extension of the *k-NN* method, which takes into account the taxonomic similarity between nearest neighbors. They report a maximum overall accuracy of 43.2%. Lendvai (2005) identifies taxonomic relations between two sections of a medical document using memory-based learning. Binary vectors represent overlap between the two sections, and the tests are run on parts of two Dutch medical encyclopedias. A best overall accuracy value of 88% is reported. Witschel (2005) proposes a methodology for extending lexical taxonomies by first identifying domain-specific concepts, then calculating semantic similarities between concepts, and finally using decision trees to insert new concepts to the right position in the taxonomy tree. The classifier is evaluated against two subtrees from GermaNet. Navigli and Velardi (2004) interpret semantically the set of complex terms that they extract, based on simple string inclusion. They make use of a variety of external resources in order to generate a semantic graph of senses. Another approach that makes use of external hierarchically structured textual resources is (Makagonov et al., 2005). The authors map an already existing hierarchical structure of technical documents to the structure of a domain-specific technical ontology. Words are clustered into concepts, and concepts into topics. They evaluate their ontology against the structure of existing textbooks in the given domain. Maedche and Volz (2001) make use of clustering, as well as pattern-based (regular expressions) approaches in order to extract taxonomies from domain-specific German texts. Degeratu and Hatzivassiloglou (2004) also make use of syntactic patterns to extract hierarchical relations, and measure the dissimilarity between the attributes of the terms using the Lance and Williams coefficient. They evaluate their methodology on a collection of forms provided by the state agencies and report a precision value of 73% and 85% for *is-a* and attributive relations respectively.

Compared to previous approaches, the work described in this chapter includes some interesting novel aspects. The whole process is based on the effort to utilize as limited external linguistic resources as possible, in order to render the methodology easily portable to other languages and other thematic domains. To this purpose no semantic networks like WordNet, grammars, hierarchically structured corpora, or pre-existing Ontologies are utilized, only two unstructured corpora of free Modern Greek text: one balanced in domain and genre, and one domain-specific.

Another interesting aspect of the present work is the language itself. Modern Greek is a relatively free-word-order language, i.e. the ordering of the constituents of a sentence is not strictly fixed, like it is in English. Therefore, it is primarily the rich morphology and not the position of a word in a sentence that determines its syntactic and semantic role. As a result, the extraction of compound terms, as well as the identification of nested terms, are not straightforward and cannot be treated as cases of simple string concatenation. The grammatical case of nouns and adjectives affects their semantic labelling. Still, the language-dependent features of the process are not so binding to not allow it to be applicable to other inflectional languages with relative easiness.

Looking at each stage of the process in more detail, there are further application-specific interesting features to be noted. As mentioned earlier in this section, classical approaches to named-entity recognition are limited to names of organizations, persons and locations. The semantic entities in the present work, however, also cover names of stocks and bonds, as well as names of newspapers (due to the newswire genre of the used corpus). Furthermore, there are other semantic types that are important for economic information retrieval, like quantitative units (e.g. denoting stock and fund quantities, monetary amounts, stock values), percentages etc. Temporal words and expressions are also identified due to their importance for data mining tasks.

Traditionally, approaches to terminology extraction make use of a domain-specific corpus that is to a large extent restricted in the vocabulary it contains and in the variety of syntactic structures it presents. The economic corpus in this work does not consist of syntactically standardized taglines of economic news. On the contrary, it presents a very rich variety in vocabulary, syntactic formulations, idiomatic expressions, sentence length, making the process of term extraction an interesting challenge.

Finally, regarding semantic relation learning, related work focuses mostly on hyperonymy/hyponymy and meronymy, in the process described here *attribute* relations are also detected, i.e. more 'abstract' relations that are specific to the economic domain. For example, *rise* and *drop* are two attributes of the concept *value*, a *stockholder* is an attribute of the concept *company*.

3. Advanced learning schemata

The lack of sophisticated resources leads unavoidably to the presence of noise in the data. Noise is examples of useless data that not only do not help the learning of useful, interesting linguistic information, but they also mislead the learning algorithm, harming its performance. In machine learning terms, noise appears in the form of class imbalance. Positive class instances (instances of the class of interest that needs to be learned) in the data are underrepresented compared to negative instances (null class instances). Class imbalance has been dealt with in previous work in various ways: oversampling of the minority class until it consists of as many examples as the majority class (Japkowicz, 2000), undersampling of the majority class (random or focused), the use of cost-sensitive classifiers (Domingos, 1999), the ROC convex hull method (Provost & Fawcett, 2001).

3.1 One-sided sampling

In the present methodology, One-sided sampling (Kubat & Matwin, 1997; Laurikkala, 2001) has been chosen to deal with the noise when learning taxonomy relations as it generally leads to better classification performance than oversampling, and it avoids the problem of arbitrarily assigning initial costs to instances that arises with cost-sensitive classifiers. One-sided sampling prunes out redundant and misleading negative examples while keeping all the positive examples. Instances of the majority class can be categorized into four groups:

Noisy are instances that appear within a cluster of examples of the opposite class; *borderline* are instances close to the boundary region between two classes; *redundant* are instances that can be already described by other examples of the same class; *safe* are instances crucial for determining the class. Instances belonging to one of the first three groups need to be eliminated as they do not contribute to class prediction. Noisy and borderline examples can be detected using *Tomek links*: two examples, *x* and *y*, of opposite classes have a distance of *δ(x,y)*. This pair of instances constitutes a Tomek link if no other example *z* exists, such that *δ(x,z)* < *δ(x,y)* or *δ(y,z)* < *δ(x,y)*. Redundant instances may be removed by creating a *consistent subset* of the initial training set. A subset C of training set T is consistent with T, if, when using the nearest neighbor (*1-NN*) algorithm, it correctly classifies all the instances in T. To this end we start with a subset C of the initial dataset T, consisting of all positive examples and a few (e.g. 20) negative examples. We train a learner with C and try to classify the rest of the instances of the initial training set. All misclassified instances are added to C, which is the final reduced dataset. The normalized Euclidean distance function is used to detect noisy and borderline examples. One-sided sampling has been used in the past in several domains such as image processing (Kubat & Matwin, 1997), medicine (Laurikkala, 2001), text categorization (Lewis & Gale, 1994).

3.2 Ensemble learning

Ensemble learning schemata have also been experimented with to deal with the noise and help the learner to disregard the useless foggy examples and focus on the useful content data. An ensemble of classifiers is a set of individual (base) classifiers whose output is combined in order to classify new instances. The construction of good ensembles of classifiers is one of the most active areas of research in supervised learning, aiming mainly at discovering ensembles that are more accurate than the individual classifiers that make them up (Dietterich, 2002). Various schemes have been proposed for combining the predictions of the base classifiers into a unique output. The most important are *bagging*, *boosting* and *stacking*. *Bagging* entails the random partitioning of the dataset in equally sized subsets (bags) using resampling (Breiman, 1996). Each subset trains the same base classifier and produces a classification model (hypothesis). The class of every new test instance is predicted by every model, and the class label with the majority vote is assigned to the test instance. Unlike bagging, where the models are created separately, *boosting* works iteratively, i.e. each new model is influenced by the performance of those built previously (Freund & Schapire, 1996; Schapire et al., 2002). In other words, new models are forced, by appropriate weighting, to focus on instances that have been handled incorrectly by older ones. Finally, *stacking* usually combines the models created by different base classifiers, unlike bagging and stacking where all base models are constructed by the same classifier (Dietterich, 2002). After constructing the different base models, a new instance is fed into them, and each model predicts a class label. These predictions form the input to another, higher-level classifier (the so-called *meta-learner*), that combines them into a final prediction.

4. The corpora

The corpora used in our experiments were:

1. The ILSP/ELEFTHEROTYPIA (Hatzigeorgiu et al., 2000) and ESPRIT 860 (Partners of ESPRIT-291/820, 1986) Corpora (a total of 300,000 words). Both these corpora are

balanced in genre and domain and manually annotated with complete morphological information. Further (phrase structure) information is obtained automatically.

2. The DELOS Corpus (Kermanidis et al., 2002) is a collection of economic domain texts of approximately five million words and of varying genre. It has been automatically annotated from the ground up. Morphological tagging on DELOS was performed by the analyzer of (Sgarbas et al., 2000). Accuracy in part-of-speech and case tagging reaches 98% and 94% accuracy respectively. Further (phrase structure) information is again obtained automatically.

All of the above corpora (including DELOS) are collections of newspaper and journal articles. More specifically, regarding DELOS, the collection consists of texts taken from the financial newspaper EXPRESS, reports from the Foundation for Economic and Industrial Research, research papers from the Athens University of Economics and several reports from the Bank of Greece. The documents are of varying genre like press reportage, news, articles, interviews and scientific studies and cover all the basic areas of the economic domain, i.e. microeconomics, macroeconomics, international economics, finance, business administration, economic history, economic law, public economics etc. Therefore, it presents richness in vocabulary, in linguistic structure, in the use of idiomatic expressions and colloquialisms, which is not encountered in the highly domain- and language-restricted texts used normally for term extraction (e.g. medical records, technical articles, tourist site descriptions). To indicate the linguistic complexity of the corpus, we mention that the length of noun phrases varies from 1 to 53 word tokens.

All the corpora have been phrase-analyzed by the chunker described in detail in (Stamatatos et al., 2000). Noun, verb, prepositional, adverbial phrases and conjunctions are detected via multi-pass parsing. From the above phrases, noun and prepositional phrases only are taken into account for the present task, as they are the only types of phrases that may include terms. Regarding the phrases of interest, precision and recall reach 85.6% and 94.5% for noun phrases, and 99.1% and 93.9% for prepositional phrases respectively. The robustness of the chunker and its independence on extravagant information makes it suitable to deal with a style-varying and complicated in linguistic structure corpus like DELOS.

It should be noted that phrases are non-overlapping. Embedded phrased are flatly split into distinct phrases. Nominal modifiers in the genitive case are included in the same phrase with the noun they modify; nouns joined by a coordinating conjunction are grouped into one phrase. The chunker identifies basic phrase constructions during the first passes (e.g. adjective-nouns, article nouns), and combines smaller phrases into lon ger ones in later passes (e.g. coordination, inclusion of genitive modifiers, compound phrases). As a result, named entities, proper nouns, compound nominal constructions are identified during chunking among the rest of the noun phrases.

5. Learning semantic entities

The tagging of semantic entities in written text is an important subtask for information retrieval and data mining and refers to the task of identifying the entities and assigning them to the appropriate semantic category. In the present work, each token in the economic corpus constitutes a candidate semantic entity. Each candidate entity is represented by a feature-value vector, suitable for learning. The features forming the vector are:

1. The token lemma. In the case where automatic lemmatization was not able to produce the token lemma, the token itself is the value of this feature.
2. The part-of-speech category of the token.

3. The morphological tag of the token. The morphological tag is a string of 3 characters encoding the case, number, and gender of the token, if it is nominal (noun, adjective or article).
4. The case tag of the token. The case tag is one of three characters denoting the token case.
5. Capitalization. A Boolean feature encodes whether the first letter of the token is capitalized or not.

For each candidate entity, context information was included in the feature-value vector, by taking into account the two tokens preceding and the two tokens following it. Each of these tokens was represented in the vector by the five features described above. As a result, a total of 25 (5x5) features are used to form the instance vectors.

The class label assigns a semantic tag to each candidate token. These tags represent the entity boundaries (whether the candidate token is the start, the end or inside an entity) as well as the semantic identity of the token. A total of 40,000 tokens were manually tagged with their class value. Table 1 shows the various values of the class feature, as well as their frequency among the total number of tokens.

Tag	Description	Percentage
AE	Start of company/organization/bank name	1.4%
ME	Middle of company/organization/bank name	0.74%
TE	End of company/organization/bank name	1.4%
E	Company/organization/bank 1-word name	1.1%
AP	Start of monetary amount/price/value	0.88%
MP	Middle of monetary amount/price/value	0.63%
TP	End of monetary amount/price/value	0.88%
AAM	Start of number of stocks/bonds	0.3%
MAM	Middle of number of stocks/bonds	0.42%
TAM	End of number of stocks/bonds	0.3%
AT	Start of percentage value	0.73%
MT	Middle of percentage value	0.08%
TT	End of percentage value	0.73%
AX	Start of temporal expression	1%
MX	Middle of temporal expression	0.75%
TX	End of temporal expression	1%
X	1-word temporal expression	0.55%
AO	Start of stock/bond name	0.16%
MO	Middle of stock/bond name	0.17%
TO	End of stock/bond name	0.16%
ON	1-word stock/bond name	0.05%
AL	Start of location name	0.21%
ML	Middle of location name	0.48%
TL	End of location name	0.21%
L	1-word location name	0.33%
F	1-word newspaper/journal name	0.14%
AN	Start of person name	0.18%
MN	Middle of person name	0.02%
TN	End of person name	0.18%
N	1-word person name	0.06%

Table 1. Values of the semantic entities class label

Unlike most previous approaches that focus on labelling three or four semantic categories of named entities, the present work deals with a total of 30 class values plus the non-entity (NULL) value, as can be seen in the previous table.

Another important piece of information provided disclosed by the previous table is the imbalance between the populations of the positive instances (entities) in the dataset, that form only 15% of the total number of instances, and the negative instances (non-entities). This imbalance leads to serious classification problems when trying to classify instances that belong to one of the minority classes (Kubat & Matwin, 1997). By removing negative examples, so that their number reaches that of the positive examples (Laurikkala, 2001), the imbalance is attacked and the results prove that classification accuracy of the positive instances improves considerably.

5.1 Experimental setup and results

Instance-based learning (*1-NN*) was the algorithm selected to classify the candidate semantic entities. *1-NN* was chosen because, due to storing all examples in memory, it is able to deal competently with exceptions and low-frequency events, which are important in language learning tasks (Daelemans et al., 1999), and are ignored by other learning algorithms.

Several experiments were conducted for determining the optimal context window size of the candidate entities. Sizes (-2, +2) - two tokens preceding and two following the candidate entity - and (-1, +1) - one token preceding and one following the candidate entity - were experimented with, and comparative performance results were obtained. When decreasing the size from (-2, +2) to (-1, +1), the number of features forming the instance vectors drops from 25 to 15. The results are shown in the second and third column of Table 2.

Another set of experiments focused on comparing classification in one stage and in two stages, i.e. stacking. In the first stage, the Instance-based learner predicts the class labels of the test instances. In the second stage, the predictions of the first phase are added to the set of features that are described in the previous section. The total number of features in the second stage, when experimenting with the (-2, +2) context window, is 30. The results of learning in two stages with window size (-1, +1) are shown in the fourth column of Table 2.

Comparative experiments were also performed with and without the removal of negative examples, in order to prove the increase in performance after applying random undersampling to the data. With random undersampling, random instances of the majority class are removed from the dataset in order for their number to reach that of the positive classes. The classification results, after applying the undersampling procedure and for context window size (-1, +1), are presented in the last column of Table 2. Testing of the algorithm was performed using 10-fold cross validation.

For a qualitative analysis of the results, a set of graphs follows that groups them together into clusters. Fig. 2 shows the impact of the selected context window size on the classification process to the various classes in the initial dataset. The bars represent the average f-score for every semantic entity type, e.g. *Stock/bond name* is the average value of the AO, MO, TO and ON classes. Certain types of entities require a larger window for their accurate detection, while larger context is misleading for other types. To the former category belong multi-word entities like stock names, person and location names. Entities that consist normally of two words at the most, or one word and a symbol (like amounts, prices, etc.) belong to the second category.

Fig. 3 shows the grouped results for the start, middle, end and 1–word labels in the initial dataset. For example, the Start bar is the average f-score over all the start labels. The Middle class group presents the lowest results, especially when a small context window size is used. This may be attributed to the fact that tokens in the inside of an entity are normally neither preceded nor followed by characteristic keywords or symbols. Therefore, their detection is harder than that of the entity borders, as the environment surrounding the entity helps the classification decision for the borders.

Class	F-score (-1,+1)	F-score (-2,+2)	F-score Stacking	F-score Undersampling
NULL	0.969	0.96	0.981	0.939
AE	0.728	0.683	0.882	0.899
ME	0.557	0.64	0.831	0.808
TE	0.768	0.74	0.871	0.903
AP	0.851	0.767	0.96	0.96
MP	0.865	0.852	0.957	0.963
TP	0.84	0.774	0.932	0.932
E	0.667	0.621	0.721	0.803
AAM	0.754	0.675	0.895	0.895
MAM	0.769	0.708	0.944	0.911
TAM	0.611	0.643	0.865	0.838
AO	0.353	0.465	0.81	0.85
MO	0.194	0.293	0.55	0.5
TO	0.143	0.35	0.629	0.611
AT	0.911	0.802	0.985	0.98
MT	0.588	0.857	0.952	0.952
TT	0.939	0.818	0.954	0.96
AX	0.585	0.558	0.755	0.806
TX	0.588	0.492	0.736	0.774
AL	0.421	0.449	0.651	0.571
ML	0.059	0.17	0.562	0.632
TL	0.278	0.293	0.524	0.465
X	0.452	0.457	0.567	0.694
F	0.889	0.947	0.944	1
AN	0.286	0.364	0.65	0.756
TN	0.378	0.632	0.65	0.579
MX	0.524	0.561	0.802	0.8
MN	0	0	0	0
ON	0	0	0	0
N	0.667	0.571	0.533	0.571
L	0.519	0.506	0.55	0.565

Table 2. Detailed experimental results

As can be seen in Table 2, classification for certain types reaches a poor score. Looking more closely at Table 1, this can be attributed without a doubt to the sparseness that characterizes

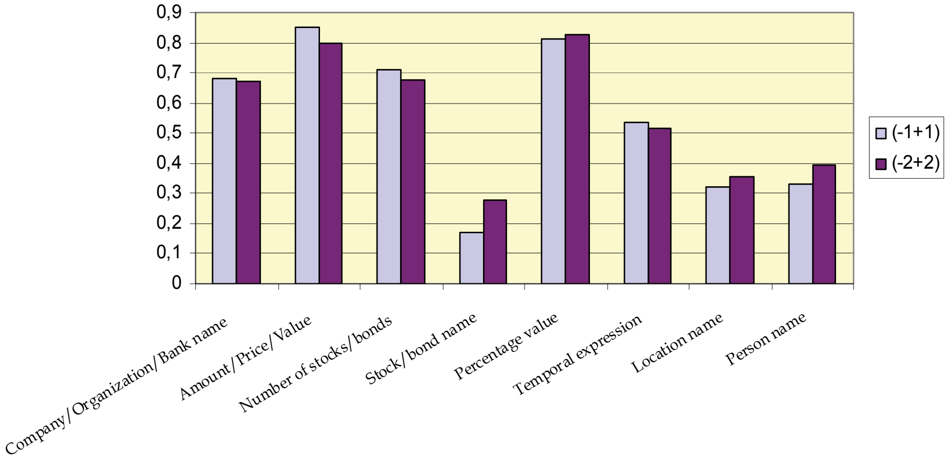

Fig. 2. The impact of the context window size

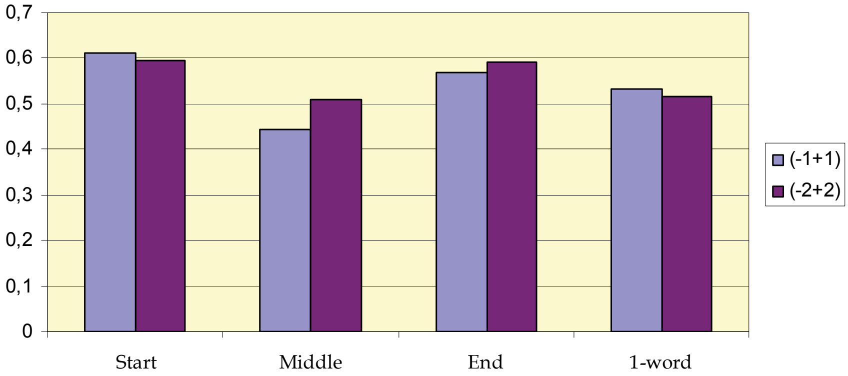

Fig. 3. The average results for the Start, Middle, End and 1-word class groups

these types (multi-word person names, multi-word stock/bond names, multi-word locations). An interesting exception to this rule is newspaper/journal names, that reach very high scores, despite their low frequency, because they are normally introduced by specific words like *'εφημερίδα'* (newspaper) or *'περιοδικό'* (journal).

Table 2 also shows the high f-score achieved for the negative (NULL) class compared to that of the positive classes, due to its high over-representation in the dataset.

The fourth column of Table 2 shows the positive effects of stacking on the task at hand. The f-score increases up to more than 50% after applying two-phase learning. This improvement is due to two reasons: first, the sequential nature of the class label tags (start, middle, end). The class of one entity depends largely on the class of the preceding and the following entities. Second, the inclusion of the predicted class of the candidate entity (from the previous learning stage) in the feature vector of the second stage forces the classifier to focus on the mistakes it made, and try to correct them. Difficult cases like multi-word locations and multi-word names are now dealt with satisfactorily.

Random undersampling also proved highly beneficial for the majority of the entity categories. It forces the learner to pay more attention to the minority classes. The random nature of the undersampling process is the reason that the results for certain entity types were not improved, as certain useful negative examples may have been removed.

The positive effects of stacking and undersampling are shown clearly in Figure 4.

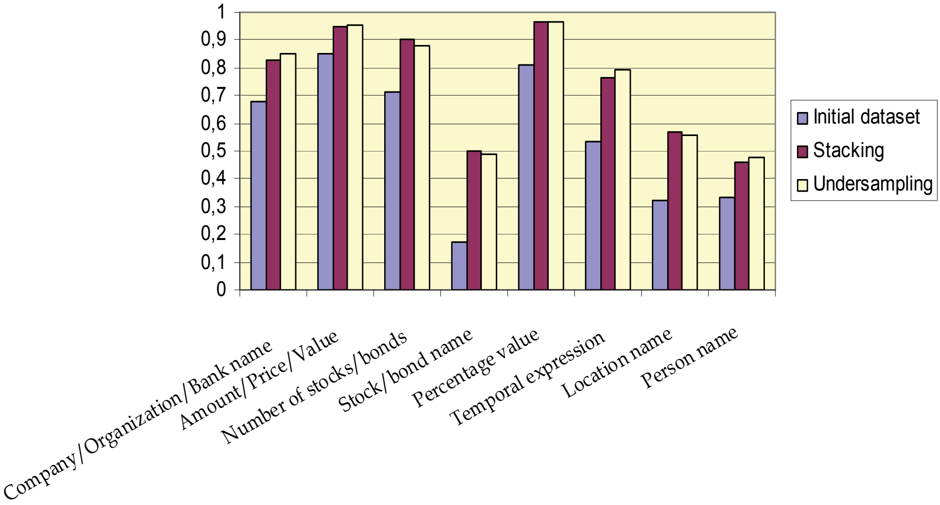

Fig. 4. The average results for all semantic entity types using Stacking and Undersampling.

One-word stock/bond names (ON) occur extremely seldom in the corpus. Person names consisting of more than two words (MN), are even more rare. The learner has not been able to detect these classes due to the sparseness.

Given, however, the nature and complexity of the corpus, the low level of pre-processing (compared to previous approaches that use phrase-chunked input), and the large number of class labels, the results of Table 2 are very impressive when compared to the ones reported in the literature.

6. Extracting economic terms

The next step of the procedure is the automatic extraction of economic terms, following the methodology described in (Thanopoulos et al., 2006). Corpora comparison was employed for the extraction of economic terms. Corpora comparison detects the difference in statistical behavior that a term presents in a balanced and in a domain-specific corpus.

Noun and prepositional phrases of the two corpora are selected to constitute candidate terms, as only these phrase types are likely to contain terms. The occurrences of words and multi-word units (n-grams), pure as well as nested, are counted. Longer candidate terms are split into smaller units (tri-grams into bi-grams and uni-grams, bi-grams into uni-grams).

Due to the relative freedom in the word ordering in Modern Greek sentences, bi-gram A B (A and B being the two lemmata forming the bi-gram) is considered to be identical to bi-gram B A, if the bi-gram is not a semantic entity. Their joint count in the corpora is calculated and taken into account. The resulting uni-grams and bi-grams are the candidate terms. The candidate term counts in the corpora are then used in statistical filters.

Statistical filtering is performed in two stages: First the relative frequencies are calculated for each candidate term. Then, for those candidate terms that present a relative frequency value greater than 1, the Log Likelihood ratio (LLR) is calculated. The LLR metric detects how surprising (or not) it is for a candidate term to appear in the domain-specific or in the balanced corpus (compared to its expected appearance count), and therefore constitute an economic domain term (or not).

Rank	Word	Translation	Count 1	Count 2	RF	LLR
1	εταιρία	company	5396	0	1845.9	852.0
2	δρχ	drachmas	3003	1	342.5	465.5
3	μετοχή	stock	2827	6	74.4	414.0
4	αγορά	buy	2330	33	11.9	257.2
5	αύξηση	growth, rise	2746	66	7.1	247.6
6	κέρδος	profit	1820	15	20.1	228.2
7	τράπεζα	bank	1367	11	20.3	171.8
8	επιχείρηση	enterprise	1969	56	6.0	162.1
9	κεφάλαιο	capital	1325	14	15.6	157.3
10	σημαντικός	important	1872	56	5.7	149.3
11	πώληση	sale	1203	11	17.9	147.3
12	προϊόν	product	1282	16	13.3	146.0
13	όμιλος	company, group	1036	5	32.2	140.0
14	Α.Ε.	INC	820	0	280.7	126. 4
15	μετοχικός	stocking	790	2	54.1	112.8
16	τιμή	price	1722	70	4.2	110.9
17	επιτόκιο	interest	821	4	31.2	110.0
18	υψηλός	high	711	0	243.4	109.2
19	κόστος	cost	1031	19	9.0	103.4
20	κλάδος	branch	833	7	19.0	103.2

Table 3. The 20 most highly ranked terms

Table 3 shows the relative frequency (*RF*) and *LLR* scores of the 20 most highly ranked economic terms, ordered by their *LLR* value. *Count 1* and *Count 2* are the term counts in the domain-specific and the balanced corpus respectively. An interesting term is '*υψηλός*', the ancient Greek form for '*high*', used today almost exclusively in the context of the degree of performance, growth, rise, profit, cost, drop (i.e. the appropriate form in economic context), as opposed to its modern form '*ψηλός*', which is used in the concept of the degree of actual height.

A particularity of the present work is that, unlike in most previous approaches to term extraction, the domain-specific corpus available to us is quite large compared to the balanced corpus. As a result, several terms that appear in DELOS do not appear in the balanced corpus, making it impossible for the LLR statistic to detect them. In other words, these terms cannot be identified by traditional corpora comparison. Lidstone's law (Manning & Schuetze, 1999) was applied to the candidate terms, i.e. each candidate term count was augmented by a value of λ=0.5 in both corpora. Thereby, terms that actually do not appear in the balanced corpus at all, end up having a *Count 2* = 0.5. This value was chosen for λ because, due to the small size of the balanced corpus, the probability of coming across a previously unseen word is significant.

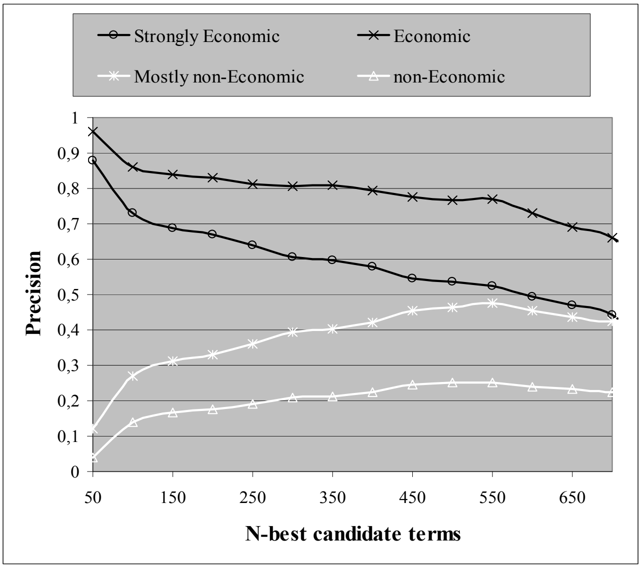

Fig. 5. Precision (y-axis) for the N-best candidate terms (x-axis) that appear in both corpora

As can be seen in Fig. 5, the term extraction methodology reaches a precision of 82% for the 200 N-best candidate terms. In this figure, *strongly economic* are terms that are characteristic of the domain and necessary for understanding domain texts. *Economic* are terms that function as economic within a context of this domain, but may also have a different meaning outside this domain. *Mostly non-economic* are words that are connected to the specific domain only indirectly, or more general terms that normally appear outside the economic domain, but may carry an economic sense in certain limited cases. *Non-economic* are terms that never appear in an economic sense or can be related to the domain in any way.

7. Learning semantic relations

The final step of the proposed methodology focuses on the identification of the taxonomic relations between the terms that were extracted in the previous phase. From the previous phase, the 250 most highly ranked terms (according to the LLR metric) were selected, and each one was paired with the rest. Syntactic and semantic information regarding the term pair has been encoded in a set of attributes that form a feature-value vector for each pair of terms. The proposed syntactic/semantic attributes are empirical and are described in the next sections. The term lemmata, their frequencies, and their part-of-speech tags were also included in the feature set. The semantic relations of a total of 6000 term pairs were manually annotated by economy and finance experts with one of the four class label values: *is-a*, *part-of*, *attribute relation* and no relation (*null*).

7.1 Semantic context vectors

The sense of a term is strongly linked to the context the term appears in. To this end, for each extracted term semantic context vectors have been constructed, that are comprised by the ten most frequent words the term co occurs with in the domain-specific corpus. A context window of two words preceding and two words following the term for every occurrence of the term in the corpus is formed. All non-content words (prepositions, articles, pronouns, particles, conjunctions) are disregarded, while acronyms, abbreviations, and certain symbols (e.g. %, €) are taken into account because of their importance for determining the semantic profile of the term in the given domain. Bi-grams (pairs of the term with each word within the con-text window) are generated and their frequency is recorded. The ten words that present the highest bi-gram frequency scores are chosen to form the context vector of the term.

7.2 Semantic similarity

For each pair of terms, their semantic similarity is calculated, based on their semantic context vectors. The smaller the distance between the context vectors, the more similar the terms' semantics. The value of semantic similarity is an integer with a value ranging from 0 to 10, which denotes the number of common words two context vectors share.

7.3 Semantic diversity

Another important semantic feature that is taken into account is how 'diverse' the semantic properties of a term are, i.e. the number of other terms that a term shares semantic properties with. This property is important when creating taxonomic hierarchies, because, the more 'shared' the semantic behaviour of a term is, the more likely it is for the term to have a higher place in the hierarchy. The notion of 'semantic diversity' is included in the feature set by calculating the percentage of the total number of terms whose semantic similarity with the focus term (one of the two terms whose taxonomic relation is to be determined) is at least 1.

7.4 Syntactic patterns

Syntactic information, regarding the linguistic patterns that govern the co occurrence of two terms, is significant for extracting taxonomic information. For languages with a relatively strict sentence structure, like English, such patterns are easier to detect (Hearst, 1992), and their impact on taxonomy learning more straightforward.

As mentioned earlier, Modern Greek presents a larger degree of freedom in the ordering of the constituents of a sentence, due to its rich morphology and its complex declination system. This freedom makes it difficult to detect syntactic patterns, and, even if they are detected, their contribution to the present task is not that easily observable.

However, two Modern Greek syntactic schemata prove very useful for learning taxonomies. They are the attributive modification schema and the genitive modification schema. The first, known in many languages, is the pattern where (usually) an adjective modifies the following noun. The second is typical for Modern Greek, and it is formed by two nominal expressions, one of which (usually following the other) appears in the genitive case and modifies the preceding nominal, denoting possession, property, origin, quantity, quality. The following phrases show examples of the first (example 1) and the second (examples 2, 3 and 4) schemata respectively.

(1) το μετοχικό[ADJ] κεφάλαιο[NOUN]
the stock capital

(2) η κατάθεση[NOUN] επιταγής[NOUN-GEN]
the deposit check
(the deposit of the check)

(3) πρόεδρος[NOUN] του συμβουλίου[NOUN-GEN]
head the council
(head of the council)

(4) αύξηση[NOUN] του κεφαλαίου[NOUN-GEN]
increase the capital
(capital increase)

Both these schemata enclose the notion of taxonomic relations: hyponymy relations (a *check deposit* is a type of deposit, a *stock capital* is a type of capital), as well as meronymy relations (the head is part of a council). The fourth example incorporates an attribute relation. The distinction among the types of relations is not always clear. In the check deposit example, the deposit may also be considered an attribute of check, constituting thereby an attribute relation. For each pair of terms, the number of times they occur in one of the two schemata in the domain-specific corpus is calculated. This information is basically the only language-dependent feature that is included in the methodology.

7.5 Experimental setup and results

9% of the term pairs belong to the *is-a* class, 17% belong to the *attribute* class and only 0.5% belong to the *part-of* class. The instances that belong to one of the first three classes are called positive, while those that belong to the null class are called negative.

Different classifiers lead to different results. Preliminary experiments have been run using various classification algorithms. *C4.5* is Quinlan's decision tree induction algorithm without pruning (Quinlan, 1993). Decision trees were chosen because of their high representational power, which is very significant for understanding the impact of each feature on the classification accuracy, and because of the knowledge that can be extracted from the resulting tree itself. The *1-NN* instanced-based learning algorithm is chosen to constitute a reference to a baseline classification performance. *SVM* is the Support Vector Machines classifier with a linear kernel. SVM cope well with the sparse data problem, and also with noise in the data (an inevitable phenomenon due to the automatic nature of the procedure described so far). A first degree polynomial kernel function was selected and the

Sequential Minimal Optimization algorithm was chosen to train the Support Vector classifier (Platt, 1998). *BN* is a Bayesian Network classifier, using a hill climbing search algorithm, and the conditional probability tables are estimated directly from the data.

	C4.5	1-NN	Naïve Bayes	SVM	BN
Is-a	0.808	0.694	0.419	0.728	0.762
Part-of	0.4	0	0	0	0
Attribute	0.769	0.765	0.77	0.788	0.775
Null	0.938	0.904	0.892	0.907	0.917

Table 4. Class f-score for various classifiers

Table 4 shows the f-score for each class achieved when trying to classify new term pairs using 10-fold cross validation. The poor results for the part-of relation are attributed mainly to its extremely rare occurrence in the data. The economic domain is more 'abstract' and is governed to a large extent by other relation types.

To overcome this problem of performance instability among the various classifiers, the application of ensemble learning is proposed. The combination of various disagreeing classifiers leads to a resulting classifier with better overall predictions (Dietterich, 2002). Experiments have been conducted using the aforementioned classifiers in various combination schemes using bagging, boosting and stacking.

Table 5 shows the results using bagging. Experiments were run using several base classifiers and several bag sizes as a percentage of the dataset size. A 50% bag size leads to the best classification results. 50% bag size means that half of the dataset instances were randomly chosen to form the first training set, another random half is used to form the second training set etc. After repeating the process ten times (10 iterations), the datasets are used to train the same base learner. Majority voting determines the class label for the test instances. The best results are achieved with a decision tree base classifier.

	C4.5	1-NN	SVM	BN
Is-a	0.856	0.736	0.728	0.766
Part-of	0	0	0	0
Attribute	0.809	0.765	0.786	0.783
Null	0.962	0.912	0.908	0.909

Table 5. Results with bagging

Table 6 shows the results using boosting. Again, various experiments were conducted with different base learners. The best results are again obtained with a decision tree base learner. It is interesting to observe the detection of some *part-of* relations using boosting.

Table 7 shows the results with stacking. Different base classifiers were combined, and their predictions were given as input to the higher level meta-learner. The combined classifiers are the 1-NN instance based-learner, the C4.5 decision tree learner, the Naïve Bayes learner, the Bayes Network classifier and the Support Vector Machine classifier. After running experiments with several combinations, it became obvious, that the greater the number and the diversity of the base classifiers, the better the achieved results. Using the same base learner combination, numerous experiments were run to compare meta-learners (shown in Table 7). The best results are achieved using SVM as a meta-learner, but the results are very satisfactory with the other meta-learners as well. It is interesting to observe that even the simple lazy meta-learner, IB1, reaches an f-score higher than 81% for all three classes. This is attributed to the predictive power of the combination of base learners. In other words, the sophisticated base learners do all the hard work, deal with the difficult cases, and the remaining work for the meta-learner is simple.

	C4.5	1-NN	SVM	BN
Is-a	0.772	0.719	0.611	0.826
Part-of	0.286	0	0	0
Attribute	0.762	0.744	0.732	0.798
Null	0.922	0.903	0.92	0.944

Table 6. Results with boosting

Meta-learner	C4.5	1-NN	Naïve Bayes	SVM
Is-a	0.761	0.848	0.827	0.853
Part-of	0	0	0	0
Attribute	0.756	0.818	0.793	0.835
Null	0.94	0.952	0.947	0.957

Table 7. Results with stacking

A further set of experiments was performed, after applying One-sided sampling to the dataset. Approximately 9% of the negative examples were removed (37.5% of which were noisy or borderline, and the remaining 62.5% were redundant). The positive effect of balancing the dataset is clearer especially when experimenting with the 'simpler' classification algorithms (IB1or C4.5), as they are more sensitive to class distribution imbalances, compared to the more 'sophisticated' classification schemata (SVM, boosting). After balancing, both sophisticated learners are able to detect part-of relations. Table 8 shows the classification results for every class.

Meta-learner	C4.5	1-NN	Naïve Bayes	SVM
Is-a	0.805	0.776	0.781	0.789
Part-of	0	0	0.25	0.33
Attribute	0.805	0.71	0.811	0.794
Null	0.931	0.913	0.915	0.927

Table 8. Results with One-sided sampling

Comparing the results with ensemble learning (Tables 5, 6 and 7) and simple learning (Table 4), the positive impact of combining multiple classifiers into a single prediction scheme becomes apparent. Mistakes made by one single classifier are amended through the iterative process and the majority voting in bagging, through instance weighting, according to how difficult an instance is to predict, in boosting, and through combining the strengths of several distinct classifiers in stacking.

Among the several ensemble schemes, stacking achieves the highest results. As mentioned earlier, class prediction performance benefits significantly from combining different base learners, because, roughly speaking, the weaknesses of one classifier are 'overshadowed' by the strengths of another, leading to a significant improvement in overall prediction.

The *part-of* relation proves to be very problematic, even with meta-learning. This is not surprising, however, taking into account that only 0.5% of the data instances were labeled as *part-of* relations. This rare occurrence leads all learning algorithms to disregard these instances, except for the unpruned decision tree learner, either as a stand-alone classifier or as base classifier in a boosting scheme. When no pruning on the decision tree is performed, overlooking tree paths that might be important for classification is avoided, and, thereby, even very low frequency events may be taken into account.

8. Discussion and future research

This chapter described the process of extracting economic knowledge automatically from Modern Greek corpora, using statistical and supervised learning techniques. The knowledge includes semantic entities, economic terminology, and semantic taxonomic relations between the extracted terms. The presented methodology makes use of no external resources in order for it to be easily portable to other domains. The language-dependent features of the described approach are kept to a minimum, so that it can be easily adapted to other languages. The lack of sophisticated resources allows for 'noise' to penetrate the dataset, leading to an imbalance between the distribution of the positive (useful for learning) and the negative (useless and misleading) class instances. Advanced sampling and ensemble learning techniques were applied, in order to remove noisy and redundant examples of the majority class, or focus on the interesting, rare instances. Despite the use of minimal resources and the highly automated nature of the process, classification performance is very promising, compared to results reported in previous work.

The extracted relations are useful in many ways. They form a generic semantic thesaurus that can be further used in several applications. First, the knowledge is important for economy/finance experts for a better understanding and usage of domain concepts. Moreover, the thesaurus facilitates intelligent search. Looking for semantically related terms improves the quality of the search results. The same holds for information retrieval and data mining applications. Intelligent question/answering systems that take into account terms that are semantically related to the terms appearing in queries return information that is more relevant, more accurate and more complete.

The economic domain is governed by semantic relations that are characteristic of the domain (buy/sell, monetary/percentage, rise/drop relations etc.), and that have been included under the attribute relation label in this work. A more fine-grained distinction between these types of attribute relations is a challenging future research direction, providing information that is very useful for data mining applications in the particular domain.

Employing other learning algorithms, that are also able to deal with the class imbalance barrier, such as neural networks, and discovering the differences in their performance compared to the algorithms presented in this chapter, promises to be another future research challenge.

Finally, another future research perspective is building an integrated ontological thesaurus from the learned taxonomic relations. Organizing the extracted terms into a hierarchical structure, e.g. a semantic network will render the extracted knowledge even more useful.

9. References

Breiman, L. (1996). Bagging Predictors. *Machine Learning*, Vol. 24, pp. 123-140

Ciaramita, M. & Altun, Y. (2005). Named Entity Recognition in Novel Domains with External Lexical Knowledge. *Proceedings of the Workshop on Advances in Structured Learning for Text and Speech Processing* (NIPS)

Cimiano, P.; Hotho, A. & Staab., S. (2004). Comparing Conceptual, Divisive and Agglomerative Clustering for Learning Taxonomies from Text. *Proceedings of the European Conference on Artificial Intelligence* (ECAI), Valencia, Spain

Daelemans, W.; van den Bosch, A. & Zavrel, J. (1999). Forgetting Exceptions is Harmful in Language Learning. *Machine Learning*, Vol. 34, pp. 11-41

Degeratu, M. & Hatzivassiloglou, V. (2004). An Automatic Model for Constructing Domain-Specific Ontology Resources. *Proceedings of the International Conference on Language Resources and Evaluation* (LREC), pp. 2001-2004, Lisbon, Portugal

Dietterich, T. (2002). *Ensemble Learning. The Handbook of Brain Theory and Neural Networks*. Second Edition. The MIT Press, Cambridge, Massachusetts, USA

Domingos, P. (1999). Metacost: A General Method for Making Classifiers Cost-sensitive. *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, pp. 155-164, San Diego, California, USA

Drouin, P. (2004). Detection of Domain Specific Terminology Using Corpora Comparison. *Proceedings of the 4th International Conference on Language Resources and Evaluation* (LREC), pp. 79−82, Lisbon, Portugal

Faure, D. & Nedellec., C. (1998). A Corpus-based Conceptual Clustering Method for Verb Frames and Ontology. *Proceedings of the LREC Workshop on Adapting Lexical and Corpus Resources to Sublanguages and Applications,* Granada, Spain

Frantzi, K.; Ananiadou, S. & Mima, H. (2000). Automatic Recognition of Multi-word Terms: the C-value/NC-value Method. *International Journal on Digital Libraries*, Vol. 3, No. 2, pp. 117−132

Freund, Y. & Schapire, R. E. (1996). Experiments with a new boosting algorithm. *Proceedings of the International Conference on Machine Learning*, pp. 148-156, San Francisco, USA

Hatzigeorgiu, N.; Gavrilidou, M.; Piperidis, S.; Carayannis, G.; Papakostopoulou, A.; Spiliotopoulou, A.; Vacalopoulou, A.; Labropoulou, P.; Mantzari, E.; Papageorgiou, H.; & Demiros, I. (2000). Design and Implementation of the online ILSP Greek Corpus. *Proceedings of the 2nd International Conference on Language Resources and Evaluation* (LREC), pp. 1737-1742, Athens, Greece

Hearst, M. A. (1992). Automatic Acquisition of Hyponyms from Large Text Corpora. *Proceedings of the International Conference on Computational Linguistics*, pp. 539-545, Nantes, France

Hendrickx, I. & van den Bosch, A. (2003). Memory-based One-step Named-entity Recognition: Effects of Seed List Features, Classifier Stacking and Unannotated Data. *Proceedings of the 7th Conference on Computational Natural Language Learning* (CoNLL), Edmonton, Canada

Hulth, A. (2003). Improved Automatic Keyword Extraction Given More Linguistic Knowledge. *Proceedings of the International Conference on Empirical Methods in Natural Language Processing* (EMNLP), pp. 216-223, Sapporo

Japkowicz, N. (2000). The Class Imbalance Problem: Significance and Strategies. *Proceedings of the International Conference on Artificial Intelligence*, Las Vegas, USA

Kermanidis, K.; Fakotakis, N. & Kokkinakis, G. (2002). DELOS: An Automatically Tagged Economic Corpus for Modern Greek. *Proceedings of the 3rd International Conference on Language Resources and Evaluation* (LREC), pp. 93-100, Las Palmas de Gran Canaria, Spain

Kilgarriff, A. (2001). Comparing Corpora. *International Journal of Corpus Linguistics*, Vol. 6, No. 1, pp. 1-37

Kubat, M. & Matwin, S. (1997). Addressing the Curse of Imbalanced Training Sets. *Proceedings of the International Conference on Machine Learning*, pp. 179- 186, Nashville, Tennessee, USA

Laurikkala, J. (2001). Improving Identification of Difficult Small Classes by Balancing Class Distribution. *Proceedings of the 8th Conference on Artificial Intelligence in Medicine in Europe*, pp. 63-66, Cascais, Portugal

Lendvai, P. (2005). Conceptual Taxonomy Identification in Medical Documents. *Proceedings of the Second International Workshop on Knowledge Discovery and Ontologies*, pp. 31-38. Porto, Portugal

Lewis, D. & Gale, W. (1994). Training Text Classifiers by Uncertainty Sampling. *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3-12, Dublin, Ireland

Maedche, A. & Volz, R. (2001). The Ontology Extraction and Maintenance Framework Text-To-Onto. *Proceedings of the Workshop on Integrating Data Mining and Knowledge Mining,* San Jose, California, USA

Makagonov, P.; Figueroa, A. R.; Sboychakov, K. & Gelbukh, A. (2005). Learning a Domain Ontology from Hierarchically Structured Texts. *Proceedings of the 22nd International Conference on Machine Learning* (ICML), Bonn, Germany

Manning, C. & Schuetze, H. (1999). *Foundations of Statistical Natural Language Processing*, MIT Press

Navigli, R. & Velardi, P. (2004). Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites. *Computational Linguistics*, Vol. 30, No. 2, pp. 151−179, MIT Press, ISSN: 0891-2017

Partners of ESPRIT-291/860. (1986). Unification of the Word Classes of the ESPRIT Project 860. Internal Report BU-WKL-0376.

Pekar, V. & Staab, S. (2002). Taxonomy Learning –Factoring the Structure of a Taxonomy into a Semantic Classification Decision. *Proceedings of the International Conference on Computational Linguistics*, Taipei, Taiwan

Pereira, F.; Tishby, N. & Lee, L. (1993). Distributional Clustering of English Words. *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*

Platt, J. (1998). Fast Training of Support Vector Machines using Sequential Minimal Optimization. *Advances in Kernel Methods - Support Vector Learning*, B. Schoelkopf, C. Burges, & A. Smola, Eds. MIT Press.

Provost, F. & Fawcett, T. (2001). Robust classification for imprecise environments. *Machine Learning*, Vol. 42, No. 3, pp. 203-231

Quinlan, R. (1993). *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo, CA

Radu, F.; Ittycheriah A.; Jing H. & Zhang T. (2003). Named Entity Recognition through Classifier Combination. *Proceedings of the 7th Conference on Computational Natural Language Learning* (CoNLL), pp. 168-171, Edmonton, Canada

Schapire, R. E.; Rochery, M.; Rahim, M. & Gupta, N. (2002). Incorporating Prior Knowledge into Boosting. *Proceedings of the Nineteenth International Conference on Machine Learning*

Sgarbas, K.; Fakotakis, N. & Kokkinakis, G. (2000). A Straightforward Approach to Morphological Analysis and Synthesis. *Proceedings of the Workshop on Computational Lexicography and Multimedia Dictionaries* (COMLEX), pp. 31-34, Kato Achaia, Greece

Sporleder, C.; van Erp, M.; Porcelijn, T.; van den Bosch, A. & Arntzen, P. (2006). Identifying Named Entities in Text Databases from the Natural History Domain. *Proceedings of the 5th International Conference on Language Resources and Evaluation* (LREC)

Stamatatos, E.; Fakotakis, N. & Kokkinakis, G. (2000). A Practical Chunker for Unrestricted Text. *Proceedings of the Conference on Natural Language Processing* (NLP), pp. 139-150, Patras, Greece

Thanopoulos, A.; Kermanidis, K. & Fakotakis, N. (2006). Challenges in Extracting Terminology from Modern Greek Texts. *Proceedings of the Workshop on Text-based Information Retrieval*, (TIR), Riva del Garda, Italy

Tsukamoto, K.; Mitsuishi, Y. & Sassano, M. (2002). Learning with Multiple Stacking for Named Entity Recognition. *Proceedings of the 6th Conference on Natural Language Learning* (CoNLL), pp. 1-4,, Taipei, Taiwan

Witschel, H. F. (2005). Using Decision Trees and Text Mining Techniques for Extending Taxonomies. *Proceedings of the Workshop on Learning and Extending Lexical Ontologies by Using Machine Learning Methods*

Wu. C.; Jan, S.; Tsai, T. & Hsu, W. (2006). On Using Ensemble Methods for Chinese Named Entity Recognition. *Proceedings of the 5th SIGHAN Workshop on Chinese Language Processing*, pp. 142-145, Sydney, Australia