Proc. Automatic Speech Recogn. and Underst. Workshop, Cancun, Mexico, Dec. 2005.

COMBINING STATISTICAL SIMILARITY MEASURES FOR AUTOMATIC INDUCTION OF SEMANTIC CLASSES

Apostolos Pangos, Elias Iosif, Alexandros Potamianos, Eric Fosler-Lussier *

Dept. of Electronics and Computer Engineering, Technical University of Crete, Chania 73100, Greece
* Dept. of Computer Science and Engineering, Ohio State University, Columbus, OH 43210, USA
{apostolis, iosife, potam}@telecom.tuc.gr, fosler@cse.ohio-state.edu

ABSTRACT

In this paper, an unsupervised semantic class induction algorithm is proposed that is based on the principle that similarity of context implies similarity of meaning. Two semantic similarity metrics that are variations of the Vector Product distance are used in order to measure the semantic distance between words and to automatically generate semantic classes. The first metric computes "wide-context" similarity between words using a "bag-of-words" model, while the second metric computes "narrow-context" similarity using a bigram language model. A hybrid metric that is defined as the linear combination of the wide and narrow-context metrics is also proposed and evaluated. To cluster words into semantic classes an iterative clustering algorithm is used. The semantic metrics are evaluated on two corpora: a semantically heterogeneous web news domain (HR-Net) and an application-specific travel reservation corpus (ATIS). For the hybrid metric, semantic class member precision of 85% is achieved at 17% recall for the HR-Net task and precision of 85% is achieved at 55% recall for the ATIS task.

1. INTRODUCTION

Many applications dealing with textual information require classification of words into semantic classes including spoken dialogue systems, language modelling, speech understanding and machine translation applications. For example, a natural language understanding module, embedded in a computer dialogue system, demands knowledge of semantic classes, in order to extract the information from the string that was transcribed by the speech recognizer. Manual construction of semantic classes is a time consuming task and often requires expert knowledge; semantic features are also sensitive to domain changes. Clearly, an automatic or semi-automatic algorithm for extracting semantic classes from text can significantly reduce development-time in many natural language processing systems. Unsupervised induction of semantic classes is also the first step towards unsupervised learning of semantics from text, the "holy grail" of natural language processing.

Among the numerous techniques and systems that have been proposed, three major families of techniques can be distinguished: numerical, symbolic and hybrid. Numerical approaches exploit the frequential aspect of data, and use statistical techniques; meanwhile, symbolic approaches examine the structure of data. Among the numerical approaches, [13] uses a semi-automatic approach in order to cluster words according to a similarity metric, working in a domain-specific corpus, ATIS. However, the resulting classes had to be hand-revised. More recently, in [11, **?**], an automatic procedure is described that classifies words and concepts into semantic classes, according to the similarity of their lexical environment. This approach induces semantically compact classes especially for restricted domains where the expressive style is oriented towards the specific needs of the certain task. Among symbolic approaches, Text-to-Onto [10, 8, 9], deals with discovering non taxonomic relationships from text and enhancing an already defined taxonomic hierarchy. The Text-to-Onto system uses shallow parsing as a natural language module. Asium [3, 2], is able to learn semantic knowledge from text by (mostly) extracting concepts/classes and putting them into taxonomic relationships. It is a semi-automatic system, meaning that user's control is needed in the process. In [7], a system is described that constructs a domain specific ontology from text documents [14]. The documents are read, processed, and a graph-structured ontology is produced using contemporary statistical methods of information retrieval such as Boolean, extended Boolean and Vector Space approaches.

The systems described above often lack the generality and/or performance necessary to make them useful for a wide range of natural language processing applications. In this paper, we extend the work of [11] and enhance the semantic similarity metric with "wide-context" information. We show that the enhanced metric can produce high-quality clustering of words into semantic classes even for semantically heterogeneous domains such as news. It is also shown that the combination of the "narrow-context" metric proposed in [11] with the "wide-context" metric outperforms both metrics. Finally, we observe that hierarchical taxonomic relationships among the induced semantic classes can be identified.

The rest of this paper is organized as follows: In Section 2, the semantic distance metrics and the automatic class induction algorithm is briefly presented. In Section 3, a step by step description of the automatic induction algorithm and its parameters is given. Experimental results and evaluation are presented in Section 4 and

Section 5 concludes the paper.

2. THEORETICAL FRAMEWORK

We investigate an iterative procedure for automatic induction of semantic classes, consisting of two main components: a *class generator* and a *corpus parser*. The *class generator*, explores the context information of every word, calculating the similarity between words; the semantic similarity distance combines two variations of the Vector Product similarity metric. Semantically similar words or concepts are grouped together into classes. The *corpus parser*, re-parses the corpus using the class definitions generated by the *class generator*, i.e., substitutes all instances of each class member with the corresponding class label. The *class generator* and *corpus parser* are run sequentially and iteratively over the corpus.

Next, we present the class generation algorithm. The two Vector Product similarity metrics are presented first: one metric computes "wide-context" similarity between words using a "bag-of-words" model and a second metric computes "narrow-context" similarity using a bigram language model [11]. Then, a linear combination of the two metrics is proposed. This section concludes with the presentation of the algorithm for merging words into classes.

2.1. Vector Product Similarity in "Bag-of-Words" Model

"Bag-of-words" or Vector Space [12] representation models view each word as one dimension in a high-dimensional vector space. Every document is considered as a vector in this space, with a zero value for every word that does not appear in the document, and a non-zero value for every word that appears in it. The non-zero values are set by using various weighting schemes — for example words that occur frequently in a document are given higher values. Similarity between two documents is measured using the cosine of the (normalized) vectors representing the two documents.

In our work, the Vector Space model is used to calculate the similarity between words that appear in a corpus of documents. The key assumption is that the context surrounding a given word provides information about its meaning. First, a vocabulary $V = (v_1, v_2, ..., v_N)$ is built containing the N unique words in the corpus. Then, a context window size WS is selected; for each word w in the vocabulary all right and left contexts of length WS are identified in the training corpus, e.g., corpus segments

$$w_{WS,L} \ ... \ w_{2,L} \ w_{1,L} \ \mathrm{w} \ w_{1,R} \ w_{2,R} \ ... \ w_{WS,R}$$

where $w_{i,L}$ and $w_{i,R}$ represent the i^{th} word to the left and to the right of w respectively. We define the left-right context set of words $N_{LR,WS}$ as the set of unique words that are found in the left and right contexts of w for a fixed context window size WS.

The feature vector for every word w is defined as $T_{w,WS} = (t_1, t_2, ..., t_N)$ where t_i is a non-negative integer and WS is the context window size. Note that the feature vector size is equal to the vocabulary size N, i.e., we have a feature for each word in the vocabulary. The i^{th} feature value t_i reflects the occurrences of vocabulary word v_i within the left or right context window WS. These non-negative values, t_i, are set according to one of two different weighting schemes:
Binary Weighting Scheme 'B':

$$t_i^B = \begin{cases} 1, & \text{if} \quad v_i \in N_{LR,WS} \\ 0, & \text{if} \quad v_i \notin N_{LR,WS} \end{cases}$$

Full Weighting Scheme 'F':

$$t_i^F = \begin{cases} c(v_i|WS), & \text{if} \quad v_i \in N_{LR,WS} \\ 0, & \text{if} \quad v_i \notin N_{LR,WS} \end{cases}$$

where $c(v_i|WS)$ is the number of occurrences of v_i within a left and right window context of size WS for the word w and for the whole corpus.

The similarity of two words, w and w', is measured as the cosine distance of their corresponding feature vectors, $T_{w,WS}$ and $T_{w',WS}$. In case of weighting scheme 'B' the similarity, VP_{1B}, is defined as:

$$VP_{1B}(w, w') = \frac{\sum_{i=1}^{N} t_i^B t_i'^B}{\sqrt{\sum_{i=1}^{N}(t_i^B)^2}\sqrt{\sum_{i=1}^{N}(t_i'^B)^2}} \qquad (1)$$

for $T_{w,WS}^B = (t_1^B, t_2^B, ..., t_N^B)$ and $T_{w',WS}^B = (t_1'^B, t_2'^B, ..., t_N'^B)$. In case of weighting scheme 'F', the similarity,VP_{1F}, is calculated using the following equation:

$$VP_{1F}(w, w') = \frac{\sum_{i=1}^{N} t_i^F t_i'^F}{\sqrt{\sum_{i=1}^{N}(t_i^F)^2}\sqrt{\sum_{i=1}^{N}(t_i'^F)^2}} \qquad (2)$$

for $T_{w,WS}^F = (t_1^F, t_2^F, ..., t_N^F)$ and $T_{w',WS}^F = (t_1'^F, t_2'^F, ..., t_N'^F)$.

2.2. Vector Product Similarity in Bigram Language Model

As above, the main idea underlying our approach is that the similarity of context implies similarity of meaning. We assume that words, which are similar in contextual distribution, have a close semantic relation [11, 4]. Consider the following sentences.

A <u>strong</u> *quake* <u>measuring</u> on the Richter Scale...
A <u>strong</u> *earthquake* <u>measuring</u> according to ...

The two italicized words occur in same lexical contexts and they are indeed similar in meaning, referring to the concept of earthquake.

A word, w, is considered with its neighboring words in a sequence

$$...w_{1,L} \ \mathrm{w} \ w_{1,R}....$$

where the words in the left and right contexts are represented by $w_{1,L}$ and $w_{1,R}$ respectively.

The similarity of two words, w_1 and w_2, is estimated as the sum of the left and right context-dependent distances. This sum gives the total "distance" between the probability distributions for, w_1 and w_2 as:

$$D^{LR}(w_1, w_2) = D^L_{12} + D^L_{21} + D^R_{12} + D^R_{21} \qquad (3)$$

where D^L and D^R are the left-context and right-context distance respectively and D_{12} denotes the (possibly asymmetric) distance between w_1 and w_2 [13, 11]. Pargellis et al [11] propose four different distance metrics D, all being computed directly from the conditional probability distributions of contexts given words, e.g., see Eqs. (5), (6). In this paper, we have used the Vector Product metric. In order to calculate the semantic distance between two words, we compute the cosine distance between two feature vectors; each feature vector of a word w measures the conditional probability of all possible contexts v_i given that word $p(v_i|w)$, i.e., each vector contains bigram language model probabilities for (context, word) pairs. Given the symmetric nature of the Vector Product metric, Eq. (3) becomes:

$$VP_2 = VP^{LR}(w_1, w_2) = VP^L_{12} + VP^R_{12} \qquad (4)$$

because $VP^L_{12} \equiv VP^L_{21}$ and $VP^R_{12} \equiv VP^R_{21}$. The two terms of Eq. (4) are defined as follows [11]:

$$VP^L_{12} == \frac{\sum_{i=1}^N p^L_1(v_i|w_1) p^L_2(v_i|w_2)}{\sqrt{\sum_{i=1}^N p^L_1(v_i|w_1)^2}\sqrt{\sum_{i=1}^N p^L_2(v_i|w_2)^2}} \qquad (5)$$

$$VP^R_{12} == \frac{\sum_{i=1}^N p^R_1(v_i|w_1) p^R_2(v_i|w_2)}{\sqrt{\sum_{i=1}^N p^R_1(v_i|w_1)^2}\sqrt{\sum_{i=1}^N p^R_2(v_i|w_2)^2}} \qquad (6)$$

where $V = (v_1, v_2, ..., v_N)$ is the vocabulary set, and $p^L_1(v_i|w_1)$ is the conditional probability of word v_i preceding w_1 in the corpus given word w_1, i.e., the v_i, w_1 bigram model probability.

2.3. Linear Combination

Combining classifiers [17, 6] is a particularly useful technique for diverse corpora, such as those that involve large amount of noise or unusually high dimensional patterns. A popular and simple way of combining multiple classifiers is simple averaging of the corresponding output values. Weighted averaging has also been proposed, along with different methods of computing the proper classifier weights [16]. In our approach, we have two different forms of the Vector Product metric with the same objective but with different perspective of the input corpus: VP_1 with broad lexical scope and VP_2 that concentrates on the immediate context of each word. Our goal is to create a combined metric that takes into account both wide-context and narrow-context information. In our experiments, a weighted linear combination of the two metrics was used as follows:

$$VP_C = \lambda_1 VP_1 + \lambda_2 VP_2 \qquad (7)$$

where λ_1+λ_2=1.

2.4. Grouping Semantically Related Words

The VP_1, VP_2 and VP_C metrics output a list of pairs, ranked according to the semantic similarity of their members, from semantically similar to semantically dissimilar. Words and semantic classes (induced in previous system iterations) are valid members of such pairs. From this list that contains all possible word pairs in the corpus, one has to choose a fixed number of top ranking pairs in order to induce the next set of semantic classes. The mapping from top ranking pairs to classes was achieved using a variant of the algorithm presented in [11]. In [11], a new class label is created for each pair and the two members are assigned to the new class. However, there is no way to merge more than two lexical units at one step which may lead to a large number of hierarchically nested classes.

Extending the work of Pargellis et al, we implemented an algorithm that creates classes that are allowed to have more than two members. This algorithm examines multiple pairs and finds those pairs that have a common element. Provided that certain conditions are met, a new class label is created and the union of these pairs is assigned to this class. Assume that the pairs (A,B), (A,C), (B,D) were ranked at the upper part of the list. According to the proposed algorithm, the class (A,B,C,D) will be created. To avoid over-generalizations only pairs that are rank ordered close to each other are allowed to participate in this process. The parameter "Search Margin", SM, defines the maximum distance between two pairs (in the semantic distance rank ordered list) that are allowed to be merged in a single class. Consider the following ranked pairs

Position in List	1	2	3	4	5
Pairs	A B	B C	E F	F G	C D

where A, B, C, D, E, F, G represent candidate words or classes. For $SM = 2$ the classes (A,B,C) and (E,F,G) will be generated, while for $SM = 3$ the classes (A,B,C,D) and (E,F,G) will be generated. By adding the search margin SM constraint it was observed that the semantic homogeny of the created classes was better preserved.

3. EXPERIMENTAL PROCEDURE

3.1. Corpora

The first corpus we experimented with was the semantically heterogeneous "HR-Net" corpus that was downloaded from the Hellenic Resources Network (http://www.hri.org). Specifically, news in English from the Hellenic Radio (ERA) between 01/01/2005 and 05/11/2005 were gathered, considering every article as a single document. HTML tags were removed from each document. The number of articles in the corpus is 2,082, the total number of words is 549,660, the size of the vocabulary is 22,904 words, the average number of words per document is 264, the maximum number of words found in a document is 1,495 and the minimum 41.

The second corpus we experimented with was the domain specific ATIS corpus. This corpus contains transcribed utterances dealing with travel information. We used an experimental corpus consisting of 1,705 utterances. The total number of words is 19,197 and the size of vocabulary is 575 words.

3.2. Bigram Language Model

For the VP_2 metric, a bigram language model was built using the CMU Statistical Language Modeling toolkit [1], applying Witten-Bell discounting [5]. Since the computation of bigram probabilities in Eq. (5), (6) over V is a time consuming procedure for the HR-Net corpus, we focused only on the “seen” bigrams (bigrams that appeared in the corpus), for which no backoff weight is needed. Furthermore, we set a threshold of k “seen” bigrams for each word participating in the pair, in order to reduce computational complexity. We followed this strategy only for the HR-Net corpus by setting $k = 5$, while for the ATIS corpus we did not demand a minimum number of “seen” bigrams to compute the similarity between words.

3.3. Experiments

The proposed system works iteratively performing the following steps:

Step 1: Calculation of the VP_1 metric.
Step 2: Calculation of the VP_2 metric.
Step 3: Normalization of the VP_1 and VP_2 results using min-max normalization.
Step 4: Calculation of the hybrid VP_C metric.
Step 5: Induction of semantic classes.
Step 6: Corpus re-parsing: all occurrences of the derived class members in the corpus are substituted by the corresponding class label.
Step 7: If specified number of iteration SI is reached stop, else go to step 1.

The experimental parameters are:

Parameter 0: The choice of the weighting scheme for the “wide-context” semantic similarity metric VP_1, i.e., use VP_{1F} or VP_{1B} as defined in Section 2.1.
Parameter 1: The size of the context window WS for metric VP_1 as defined in Section 2.1.
Parameter 2: The number of system iterations (SI).
Parameter 3: The number of induced semantic classes per iteration (IC).
Parameter 4: The size of Search Margin (SM) defined in Section 2.4.
Parameter 5: The values of λ_1 and λ_2 ($= 1-\lambda_1$) defined in Eq. (7). These values were estimated on held out data in order to maximize the performance of the VP_C metric.

4. EVALUATION

In order to evaluate the induced semantic classes for the HR-Net corpus, we used as a benchmark a manually crafted semantic taxonomy. Two researchers were assigned this task, devising a taxonomy of 43 semantic classes with 1,820 word-members in them. Every word was assigned to a single class. In order to avoid infrequent words, a threshold was adopted and only words with frequency greater than 9 in the corpus were included in the taxonomy. Regarding the experimental procedure, in order to decrease computation time, our system was tested only for these 1,820 words. The following table illustrates 5 representative handcrafted classes along with example members.

Class Name	Members
Education	university, school, student...
Politics	Karamanlis, president, minister...
Law	prosecutor, judge, crime...
Health	hospital, surgery, pharmaceutical...
Sports	Olympiacos, UEFA, Rehhagel ...

For the evaluation procedure of the ATIS corpus, we used a manually crafted semantic taxonomy, consisting of 32 classes that include a total of 291 members. Every word was assigned to a single class. Regarding the experimental procedure, we generated manually characteristic chunks, like New York → New_York, J F K → J_F_K etc. Also, during the experiments on ATIS, all the words of the vocabulary were taken under consideration. The following table shows 5 representative handcrafted classes along with some members.

Class Name	Members
City	Atlanta, Dallas, Las_Vegas...
Day	Monday, Tuesday, Friday...
Fairtype	one_way, round_trip, nonstop...
Airline	Delta, Lufthansa, T_W_A...
Meal	meal, lunch, breakfast ...

Although a hierarchical semantic taxonomy was also constructed for both corpora, the evaluation focused only on the flat (terminal) semantic classes presented above. In other words, every induced class was evaluated with respect only to the corresponding handcrafted class without examining its relationships with other classes over the taxonomy. An induced class is assumed to correspond to a handcrafted class, if at least half of its members are included (“correct members”) in the handcrafted class. Precision and recall are calculated as follows:

$$\text{Precision} = \frac{\sum_{i=1}^{m} c_i}{\sum_{i=1}^{m} \alpha_i} \qquad \text{Recall} = \frac{\sum_{i=1}^{m} c_i}{\sum_{j=1}^{r} \beta_j}$$

where m is the total number of induced classes, r is the total number of handcrafted classes, c_i is the “correct members” of the i^{th} induced class, α_i is the total number of members of the i^{th} induced class and β_j is the total number of members of the j^{th} handcrafted class that appear in the corpus.

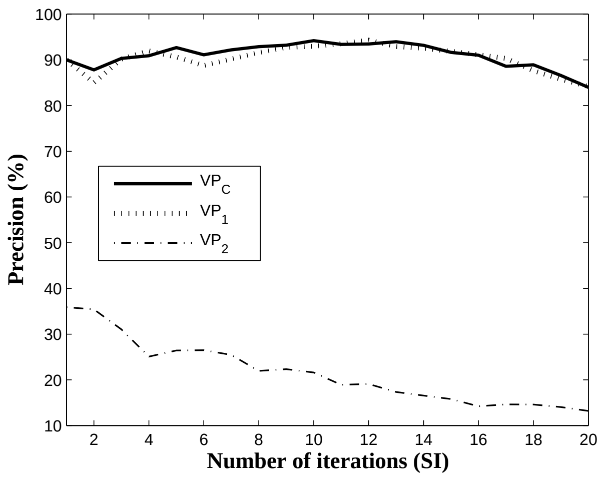

Fig. 1. Cumulative precision on the HR-Net corpus as a function of system iterations for the three metrics.

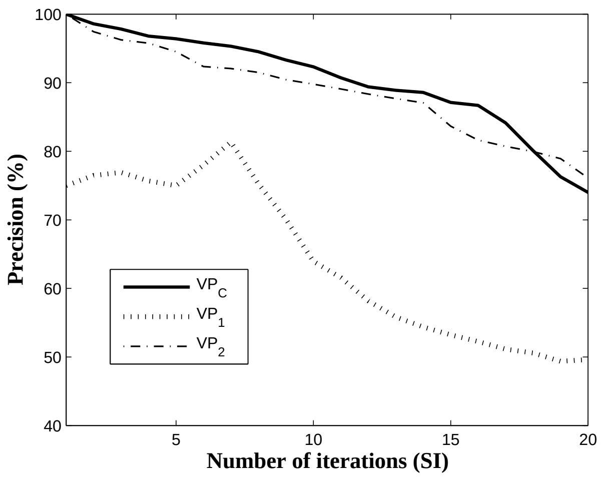

Fig. 2. Cumulative precision on the ATIS corpus as a function of system iterations for the three metrics.

4.1. Evaluation Results on the HR-Net corpus

Fig 1 illustrates the performance of the three metrics, VP_1, VP_2 and VP_C on the HR-Net corpus. The parameters used in this experiments were VP_{1B}, $WS = 100$, $SI = 20$, $IC = 10$, $SM = 10$, $\lambda_1 = 0.9$ (and $\lambda_2 = 0.1$). Clearly, the VP_1 metric significantly outperforms the VP_2 metric on this task. This is expected because the HR-Net corpus is semantically diverse and words with similar "narrow-context" are often not semantically related. Despite the poor precision of VP_2, this metric contributes to the combined metric VP_C by identifying those words that have a distinct bigram context. The linear combination of the two metrics, VP_C, tends to achieve the best results but the difference between VP_1 and VP_C is often small or non-existent. The highest precision, 94.19%, is obtained by VP_C at the end of the 10^{th} iteration.

The following table presents the cumulative values of achieved recall as a function of system's iterations and metric used (VP_C, VP_1 or VP_2) on the HR-Net corpus (same parameters as above).

	Recall		
SI	VP_C	VP_1	VP_2
5	5.54%	4.94%	3.29%
10	10.27%	9.72%	6.15%
15	14.34%	13.62%	8.62%
20	17.30%	16.86%	10.87%

It can be seen, that VP_C has slightly higher recall than VP_1 and that VP_2 is clearly the worst out of the three metrics. Note that for $SI = 20$ the VP_C metric generates 24 classes with 315 members.

4.2. Evaluation Results on the ATIS corpus

Fig. 2 shows the precision achieved by the three metrics VP_1, VP_2 and VP_C on the ATIS corpus. For this experiment we used the following parameters: VP_{1B}, $WS = 100$, $SI = 20$, $IC = 10$, $SM = 5$, $\lambda_1 = 0.2$ (and $\lambda_2 = 0.8$). On this domain-specific corpus the performance of the VP_1 and VP_2 metrics is reversed. VP_2 clearly outperforms VP_1 both in terms of precision and recall (shown in the table that follows). This is expected because in this semantically homogeneous corpus the "narrow-context" similarity often signifies semantic similarity; while there is not enough data to adequately train the statistics of the "wide-context" metric. The precision of the linear combination of the two metrics, VP_C is slightly higher than the precision of VP_2, although, by the end of the 20^{th} iteration this advantage is gone. Note that the best precision for VP_2 and VP_C (almost 100%) is achieved in the first iteration of the system.

The following table presents the cumulative values of achieved recall as a function of system iterations and metric used.

	Recall		
SI	VP_C	VP_1	VP_2
5	28.52%	7.9%	30.58%
10	40.54%	20.96%	43.29%
15	52.57%	35.05%	52.92%
20	60.48%	44.32%	59.1%

It can be seen, that VP_C and VP_2 have similar recall values, while VP_1 is significantly worse. Note that for $SI = 20$ the VP_C metric generates 27 classes with 176 members.

5. CONCLUSIONS AND FUTURE WORK

We have presented an unsupervised procedure for automatic induction of semantic classes using a "wide-context" VP_1 and a "narrow-context" VP_2 semantic similarity metric as well as their combination VP_C. It was shown that VP_1 performs best for the

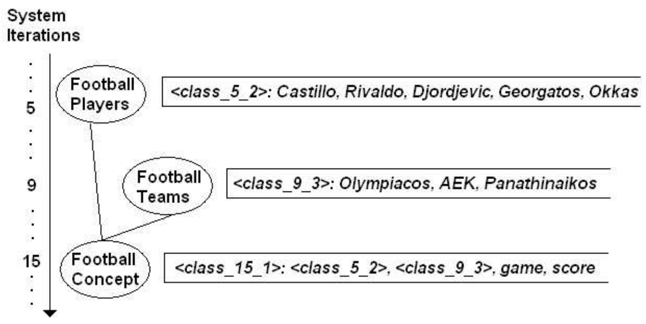
System Iterations
5
9
15
Football Players
Football Teams
Football Concept
<class_5_2>: Castillo, Rivaldo, Djordjevic, Georgatos, Okkas
<class_9_3>: Olympiacos, AEK, Panathinaikos
<class_15_1>: <class_5_2>, <class_9_3>, game, score

Fig. 3. Pictorial view of relationships among induced classes for HR-Net

semantically heterogeneous HR-Net corpus, while VP_2 performs better for the domain-specific ATIS corpus. The hybrid metric VP_C performed slightly better than the best of VP_1, VP_2 in our experiments. Semantic class precision of 85% and recall of 17% was obtained for the HR-Net corpus. Precision of 85% and recall of 55% was achieved for the ATIS corpus.

More work is needed to improve on each of the metrics and, especially, on the combined metric. During experimentation we also observed that the system generated automatically some internal relationships between previous induced semantic classes. A representative example, taken from the experiments over the HR-Net corpus is shown in Fig. 3. This aspect of the system can be considered as a promising step towards the automatic extraction of a hierarchy between taxonomic classes.

Acknowledgements

This work was partially supported by the EU-IST-FP6 MUSCLE network of excellence.

6. REFERENCES

[1] Clarkson, P.R., Rosenfeld, R., "*Statistical Language Modeling Using the CMU-Cambridge Toolkit*." In: Proc. Fifth European Conf. on Speech Comm. and Tech., 1997.

[2] Faure, D. and N'Edellec C., "*A Corpus-based Conceptual Clustering Method for Verb Frames and Ontology Acquisition*." ASIUM System, 1998.

[3] Faure, D. and N'Edellec C., "*ASIUM: Learning subcategorization frames and restrictions of selection*." 10th Conference on Machine Learning (ECML 98), Workshop on Text Mining, Chemnitz, Germany, 1998.

[4] Herbert R., Goodenough, B.J., "*Contextual Correlates of Synonymy*." Communications of the ACM, vol. 8, 1965.

[5] Jurafsky, D., Martin, J.H., "*Speech and Language Processing*." Prentice Hall. Upper Saddle River, 2000.

[6] Larkey. S. and Croft, W., "*Combining classifiers in text classification.*." In Proc. SIGIR, pp. 81-93., 1996.

[7] Maddi, G. R. and Velvadapu, C. S., "*Ontology Extraction from text documents by Singular Value Decomposition*." Bowie State University, 2001.

[8] Maedche, A. and Staab, S., "*Discovering Conceptual Relations from Text*." Technical Report 399, Institute AIFB, Karlsruhe University, 2000.

[9] Maedche, A. and Staab, S., "*The Text-to-Onto Ontology Learning Environment*." Institute AIFB, Karlsruhe University, 2000.

[10] Maedche, A. and Staab, S., "*Ontology learning for the Semantic Web*." IEEE Intelligent Systems, 16(2), 2001.

[11] Pargellis, A., Fosler-Lussier, E., Lee, C, Potamianos, A. and Tsai, A., "*Auto-Induced Semantic Classes*." Speech Communication. 43, 183-203., 2004.

[12] Sebastiani, F., "*Machine learning in automated text categorization*." ACM Computing Surveys, 34(1):1 47, 2002.

[13] Siu, K.-C., Meng, H.M., "*Semi-automatic acquisition of domain-specific semantic structures*." In: Proc. Sixth European Conf. on Speech Comm. and Tech., Budapest, vol. 5, pp 2039-2042, 1999.

[14] Srivastava, S., Lamadrid, J. G. and Karakashyan, Y., "*CADIP 2000: Document Ontology Extractor*." Bowie State University, 2000.

[15] Theodoridis, S., Koutroumbas, K., "*Pattern Recognition*." Elsevier Academic Press, 2003.

[16] Tumer, K. and Ghosh, J., "*Theoretical foundations of linear and order statistics combiners for neural pattern classifiers.*." IEEE Trans. Neural Networks, 1995.

[17] Woods, K., Kegelmeyer, W. and Bowyer Jr, K., "*Combination of multiple classifiers using local accuracy estimates.*." IEEE Trans. PAMI, 19, 405-410., 1997.