# UNSUPERVISED COMBINATION OF METRICS FOR SEMANTIC CLASS INDUCTION

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

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
{iosife, tegos, apostolis, potam}@telecom.tuc.gr, fosler@cse.ohio-state.edu

## ABSTRACT

In this paper, unsupervised algorithms for combining semantic similarity metrics are proposed for the problem of automatic class induction. The automatic class induction algorithm is based on the work of Pargellis et al [1]. The semantic similarity metrics that are evaluated and combined are based on narrow- and wide-context vector-product similarity. The metrics are combined using linear weights that are computed 'on the fly' and are updated at each iteration of the class induction algorithm. Specifically, the weight of each metric is selected to be inversely proportional to the inter-class similarity of the classes induced by that metric and for the current iteration of the algorithm. The proposed algorithms are evaluated on two corpora: a semantically heterogeneous news domain (HR-Net) and an application-specific travel reservation corpus (ATIS). It is shown, that the (unsupervised) adaptive weighting scheme outperforms the (supervised) fixed weighting scheme. Up to 50% relative error reduction is achieved by the adaptive weighting scheme.

***Index Terms***— text processing, information retrieval, ontology creation

## 1. INTRODUCTION

Many applications dealing with textual information require classification of words into semantic classes including spoken dialogue systems, language modeling, speech understanding and machine translation applications. Manual construction of semantic classes is a time consuming task and often requires expert knowledge; semantic features are also sensitive to domain changes. An automatic or semi-automatic algorithm for extracting semantic classes from text leads to the rapid development of many natural language processing systems. Semantic class construction and semantic similarity metrics are also important for web applications such as web search and document retrieval.

Among the numerous techniques and systems that have been proposed, many approaches exploit the frequential aspect of data, and use statistical techniques. A semi-automatic approach is used by [2] in order to cluster words according to a similarity metric, working in a domain-specific corpus, ATIS. In [1, 3], 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. More recently, [4] proposes an algorithm that combines "wide-context" and "narrow-context" similarity metrics using constant weights that were estimated on held-out data during an a priori experimental procedure.

In this paper, we focus on the problem of adaptive unsupervised weight estimation for combining multiple similarity metrics. The class induction system proposed in [4] works iteratively, updating a hierarchical structure of semantic classes in each iteration. Our motivation for creating an adaptive weighting scheme, is that the relative performance of each metric (in terms of precision and recall) varies from iteration to iteration. It is expected that by updating the weights of each metric at each iteration the combined metric performance can significantly improve. In addition, our goal is to create a fully unsupervised metric combination algorithm that *does not* require experiment on held-out data to compute the weights. Next, we propose a fully unsupervised adaptive algorithm for combining semantic similarity metrics. The proposed algorithm is evaluated on two tasks and it is shown to outperform (supervised) fixed weight combination schemes.

## 2. SEMANTIC CLASS INDUCTION

As in [4], we follow 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 metric combines two or more variations of the Vector Product similarity metric. Semantically similar words or concepts are grouped together into classes. The *corpus parser*, reparses 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.

### 2.1. Vector Product Similarity Metrics

Our approach relies on the idea that the similarity of context implies similarity of meaning. We assume that words, which are similar in contextual distribution, have a close semantic relation [1, 5]. Both narrow- and wide-context is taken into account as described next.

In "Bag-of-words" [6] models, for each word $w$ in the vocabulary a context window size $WS$ is selected. The right and left contexts of length $WS$ in the corpus are considered for word $w$, e.g., $[v_{WS,L}$ ... $v_{2,L}$ $v_{1,L}]$ $w$ $[v_{1,R}$ $v_{2,R}$ ... $v_{WS,R}]$, where $v_{i,L}$ and $v_{i,R}$ represent the $i^{th}$ word to the left and to the right of $w$ respectively. The feature vector for every word $w$ is defined as $T_{w,WS} = (t_{w,1}, t_{w,2}, ..., t_{w,N})$ where $t_{w,i}$ 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 $V$. The $i^{th}$ feature value $t_{w,i}$ reflects the occurrences of vocabulary word $v_i$ within the left or right context window $WS$. This feature value is set according to a Binary (Bin.) or a Term Frequency (Freq.) Scheme. Binary Scheme assigns 1 if the word $v_i$ appears within the left and right window context of size

$WS$ for the word $w$, while Term Frequency Scheme assigns the number of occurrences of $v_i$ in left and right $WS$. Both schemes assign a 0 value if $v_i$ does not exist within $WS$. The "Bag-of-words" metric, $S_{A,WS}$, using Binary or Term Frequency Scheme, measures the similarity of two words, $w_1$ and $w_2$, as the cosine distance of their corresponding feature vectors, $T_{w_1,WS}$ and $T_{w_2,WS}$ [4]:

$$S_{A,WS}(w_1, w_2) = \frac{\sum_{i=1}^{N} t_{w_1,i} t_{w_2,i}}{\sqrt{\sum_{i=1}^{N}(t_{w_1,i})^2}\sqrt{\sum_{i=1}^{N}(t_{w_2,i})^2}} \quad (1)$$

given a context window of length $WS$.

In an n-gram language model a word $w$ is considered with its neighboring words $v_{1,L}$ and $v_{1,R}$ in the left and right contexts within a sequence. In order to calculate the similarity of two words, $w_1$ and $w_2$, 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. Semantic similarity is defined as

$$S_B(w_1, w_2) = S_B^L(w_1, w_2) + S_B^R(w_1, w_2) \quad (2)$$

where the two terms of Eq. (2) are [1]:

$$S_B^L(w_1, w_2) = \frac{\sum_{i=1}^{N} p(v_{i,L}|w_1)p(v_{i,L}|w_2)}{\sqrt{\sum_{i=1}^{N} p(v_{i,L}|w_1)^2}\ \sqrt{\sum_{i=1}^{N} p(v_{i,L}|w_2)^2}} \quad (3)$$

$$S_B^R(w_1, w_2) = \frac{\sum_{i=1}^{N} p(v_{i,R}|w_1)p(v_{i,R}|w_2)}{\sqrt{\sum_{i=1}^{N} p(v_{i,R}|w_1)^2}\ \sqrt{\sum_{i=1}^{N} p(v_{i,R}|w_2)^2}} \quad (4)$$

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

### 2.2. Generating Word Classes

Using similarity metric $S_{A,WS}$ or $S_B$ the class induction system outputs a list of pairs, ranked according to the semantic similarity of their members, from semantically similar to semantically dissimilar. 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. COMBINED SIMILARITY METRICS

Combining similarity metrics of various context lengths makes it possible to utilize multiple lexical scopes of the contextual information. In [4], two different metrics were linearly combined using fixed weights: $S_{A,WS=50}$ with broad lexical scope and $S_B$ that focuses on the immediate context of each word.

Our goal is to improve on the combined metric by adding multiple context lengths and, most importantly, to automatically and adaptively estimate the weights assigned to each contributing individual metric. We propose the following weighted linear combination:

$$C(w_1, w_2) = \sum_{m=1}^{M} \lambda_m S_m(w_1, w_2) \quad (5)$$

where $S_m$ can be any of $S_{A,WS}(w_1, w_2)$ or $S_B(w_1, w_2)$ (for various context lenghts $WS$). Note that $\lambda_m$ varies from iteration to iteration and that

$$\sum_{m=1}^{M} \lambda_m = 1 \quad (6)$$

### 3.1. Unsupervised Weight Estimation

The hybrid metric $C$ takes into account different metrics with different lexical scopes using a weighted linear combination. The weight $\lambda_m$ estimation algorithm is motivated by the work in [7], where it is shown that the "optimal" weights for the combined classifier should be approximately inversely proportional to the classification error rate of each (stand-alone) classifier. In order to have an unsupervised weighting metric, however, we need some estimate of the individual classification error rates of the component metrics. Assuming equal priors (and variance normalized data) the classification error rate can be assumed to be approximately proportional to the inter-class similarity (or inversely proportional to the inter-class distance). Thus the optimal weights should be inversely proportional to the inter-class similarity. This agrees with our intuition that greater importance should be given to the individual metric $S_m$ that achieves better class separability.

Consider that (at iteration $I$) the metric $S_m$ generates a number of classes $c_{i,m}$ from the top $NP$ ranking word pairs; where $i$ is the class index for metric $S_m$. The quality of class induction for each metric at iteration $I$ is measured by employing a criterion of inter class similarity. Specifically, the inter class similarity between two classes, $c_i$ and $c_j$ generated by $S_m$ is computed as:

$$D_{i,j,m} = \frac{\sum_{w_k \in c_{i,m}} \sum_{w_l \in c_{j,m}} S_m(w_k, w_l)}{\mid c_{i,m} \mid \ \mid c_{j,m} \mid} \quad (7)$$

where $\mid . \mid$ denotes set cardinality. The average inter class similarity $D_{m,avg}$ for metric $S_m$ is computed by averaging over all similarity scores between all possible pairs of classes $(i, j)$:

$$D_{m,avg} = \langle D_{i,j,m} \rangle_{(i,j)}. \quad (8)$$

Finally, the combination weight $\lambda_m$ assigned to $S_m$ is equal to the inverse of the average inter class similarity:

$$\lambda_m = \frac{1}{\mu_m D_{m,avg}} \quad (9)$$

Note that $\mu_m$ is a smoothing factor and is a moving average of $D_{m,avg}$ over all past system iterations.

## 4. EXPERIMENTAL CORPORA AND PROCEDURE

The first corpus we experimented with was the domain specific ATIS corpus which consists of 1,705 transcribed utterances dealing with travel information. The total number of words is 19,197 and the size

of vocabulary is 575 words. The second experimental corpus was the semantically heterogeneous "HR-Net" corpus dealing with news. The total number of words is 549,660 and the size of the vocabulary is 22,904 words [4].

For the $S_B$ metric, the Bigram Language Model was built using the CMU Statistical Language Modeling toolkit, applying Witten-Bell discounting and using back-off weights to compute the probability of unseen bigrams.

Regarding the experimental steps, several variations of $S_{A,WS}$ and $S_B$ are calculated and their results are normalized using min-max normalization. Then the hybrid $C$ metric is calculated and semantic classes are induced according to the algorithm described in Section 2.2. The weights $\lambda$ are computed using Eq. (9). Finally, all occurrences of the derived class members in the corpus are substituted by the corresponding class label and the above procedure is repeated until the specified number of iteration $SI$ is reached.

The following parameters must be defined: (i) the context window $WS$ for $S_{A,WS}$ as well as the scheme used, Binary (Bin.) or Frequency (Freq.) as described in Section 2.1, (ii) the total number of system iterations ($SI$), (iii) the number of induced semantic classes per iteration ($IC$), (iv) the size of Search Margin ($SM$) defined in Section 2.2, and (v) the number of pairs $NP$ considered for inter-class similarity and weight computation as discussed in Section 3.1.

## 5. EVALUATION

In order to evaluate the induced semantic classes for the HR-Net corpus, we used as a benchmark a taxonomy of 43 semantic classes including 1,820 word-members, manually crafted by two researchers. Every word was assigned *only to one* hand-crafted class and our system was tested only for these 1,820 words. For the evaluation procedure of the ATIS corpus, we used a manually crafted semantic taxonomy, consisting of 38 classes that include a total of 308 members. Every word was assigned only to one hand-crafted class. For experimental purposes, we generated manually characteristic "word chunks", e.g., T W A $\rightarrow$ T_W_A. Also, for the ATIS experiments, all 575 words in the vocabulary were used for similarity metric computation and evaluation.

For both corpora the evaluation focused only on the terminal semantic classes (hierarchical class generation was not evaluated). 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 50% 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, $\alpha_i$ is the total number of members of the $i^{th}$ induced class and $\beta_j$ is the total number of members of the $j^{th}$ handcrafted class that occur in the corpus.

In Fig. 1(a), the cumulative precision achieved by the metrics $S_{A,WS}$, $S_B$ and their combination $\lambda_1 S_{A,WS=3} + \lambda_2 S_B$, is shown for the ATIS corpus (solid line). The weights for the combined metric are computed adaptively at each iteration using Eq. (9); the weights are shown in Fig. 1(b). For this experiment we used the following parameters: Freq. Scheme for $S_{A,WS=3}$, $SI = 20$, $IC = 10$, $SM = 5$, $NP = 50$. Note that the cumulative precision of the combined metric with fixed weights $\lambda_1 = 0.35$ and $\lambda_2 = 0.65$ is also shown in Fig. 1(a) (dotted line). It is interesting to note that the adaptive weighting schemes significantly outperforms the fixed weighting scheme in this experiment. The precision achieved for the metric $S_{A,WS=3}$ is good in the first few iterations but quickly decreases well below the precision of the $S_B$ metric. In general, the ATIS corpus favors the narrow-context metrics as also shown in [4]. The adaptive weights $\lambda_{1,2}$ in Fig 1(b) take reasonable values; as the precision of $S_{A,WS=3}$ relative to $S_B$ decreases so does $\lambda_1$ relative to $\lambda_2$. The increase in the value of $\lambda_1$ after iteration 8 could be due to sparse data for weight estimation. Overall, at iteration $SI = 20$ the adaptively weighted combined metric generates 33 classes with 224 members achieving a recall of 72.7% (see table below).

In Fig. 1(c), the cumulative precision achieved by three variations of $S_{A,WS}$ (Binary scheme, window sizes 50, 10, and 2), and their combinations is shown for the HR-Net corpus. Two combined metrics are shown both using adaptively computed weights: $C = \lambda_1 S_{A,WS=50} + \lambda_2 S_{A,WS=10} + \lambda_3 S_{A,WS=2}$, and $C' = \lambda'_1 S_{A,WS=50} + \lambda'_2 S_{A,WS=10}$. Note that: $SI = 20$, $IC = 10$, $SM = 10$, $NP = 50$ for this experiment. The three metric combination $C$ significantly outperforms the two metric combination $C'$ in terms of precision. Also note that $C$ outperforms the best of the $S_{A,WS}$ metrics and achieves up to 50% relative error rate reduction. The semantically heterogeneous nature of HR-Net corpus allows multiple metrics of different lexical scopes to be combined successfully. In Fig. 1(d), the assigned weights for the three-metric combination $C$ are shown. During the very early iterations $S_{A,WS=10}$ is weighted most, but after the $6^{th}$ iteration $S_{A,WS=50}$ is assigned the greatest weight. The metric with the smallest window size, $S_{A,WS=2}$, is given the lowest $\lambda$ value throughout[1]. This is consistent with the corresponding precision curves and agrees with our intuition. Note that at iteration $SI = 20$ the combined metric $C$ generates 20 classes with 304 members and achieves precision of 16.7%.

In the following table, which shows the recall across different experiments, we can see that the adaptive weightings are relatively close to the fixed parameters in terms of recall, so the improvement in precision is not at the expense of recall.

| Recall at SI(%) | 1 | 5 | 10 | 15 | 20 |
|---|---|---|---|---|---|
| ATIS:adaptive $\lambda_{1,2}$ | 11 | 31.2 | 46.1 | 64 | 72.7 |
| ATIS:fixed $\lambda_{1,2}$ | 12.7 | 33.1 | 49.7 | 65.3 | 75 |
| ATIS:$S_{A,WS=3}$ | 14.9 | 34.4 | 44.8 | 53.2 | 63.3 |
| ATIS:$S_B$ | 6.5 | 30.2 | 45.8 | 59.4 | 70.1 |
| HR-Net:adaptive $\lambda_{1,2,3}$ | 1.3 | 5 | 9.8 | 13.9 | 16.7 |
| HR-Net:adaptive $\lambda'_{1,2}$ | 0.8 | 5 | 9.6 | 13.3 | 16.6 |
| HR-Net:$S_{A,WS=50}$ | 1 | 5.7 | 9.5 | 13.5 | 16.9 |

## 6. CONCLUSIONS AND FUTURE WORK

We have presented an algorithm for unsupervised computation of weights to individual metrics of different lexical scopes. The metrics are linearly combined into a hybrid metric. The proposed algorithm monitors the efficiency of each individual metric and attempts to assign greater weight to the "best-performing" metric. Experiments on

[1]This is consistent with the experiments in [4] showing that the wide-context metrics outperform the narrow-context ones for the semantically heterogeneous HR-Net corpus. It is interesting to note that when the proposed unsupervised weight computation algorithm is used there is no need to select metrics based on corpus characteristics. Instead a corpus independent combined metric can be used and automatically poor performing metrics will be weighted less in the combination.

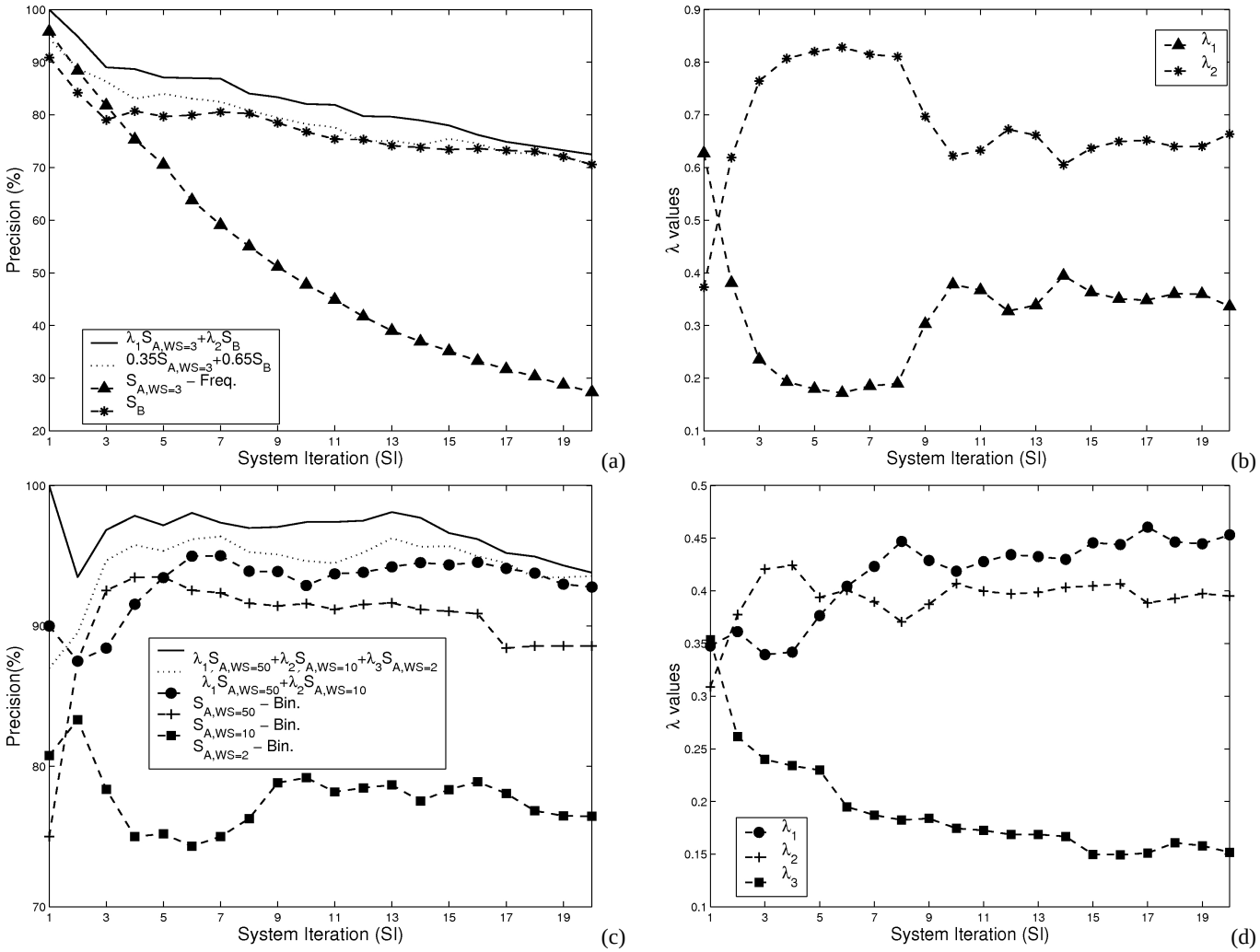


**Fig. 1**. (a) Cumulative precision for the ATIS task for two individual metrics and two combined metrics (adaptive vs fixed weighting scheme), (b) Assigned weights for the ATIS task (adaptive weighting scheme), (c) Cumulative precision for the HR-Net task for three individual and two combined metrics (three-way and two-way adaptive combination), (d) Assigned weights for the three-way combination metric for the HR-Net task.

two different corpora showed that the adaptively computed weights outperform the fixed weight computation scheme. Also three metric combination (wide-, mid- and narrow-context size) significantly outperformed each one of the individual metrics in terms of precision and recall of generated classes. The proposed unsupervised metric combination algorithm makes it possible to employ a *corpus-independent semantic similarity* metric for semantic class induction. Future work will investigate how to include estimation error variance is the weight estimation criterion as discussed in [7]. Combination of other types of semantic similarity measures will also be investigated.

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

## 7. REFERENCES

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

[2] Siu, K.-C., Meng, H.M., "Semi-automatic acquisition of domain-specific semantic structures," In: Proc. EUROSPEECH, 1999.

[3] Pargellis, A., Fosler-Lussier, E., Potamianos, A., Lee, C., "A comparison of four metrics for auto-inducing semantic classes," In: Proc. ASRU, 2001.

[4] Pangos, A., Iosif, E., Potamianos, A., Fosler-Lussier, E., "Combining statistical similarity measures for automatic induction of semantic classes," In: Proc. ASRU, 2005.

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

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

[7] Potamianos, A., Sanchez-Soto, E., Daoudi, K., "Stream weight computation for multi-stream classifiers," In: Proc. ICASSP, 2006.