LOOKING FOR TOPIC SIMILARITIES OF HIGHLY INFLECTED LANGUAGES FOR LANGUAGE MODEL ADAPTATION

Mirjam Sepesy Maučec, Zdravko Kačič, Bogomir Horvat

University of Maribor, Faculty of Electrical Engineering and Computer Science
Smetanova 17, SI-2000 Maribor, Slovenia
mirjam.sepesy@uni-mb.si

6th International Conference on Spoken Language Processing (ICSLP 2000)
Beijing, China
October 16-20, 2000

ABSTRACT

In this paper, we propose a new framework to construct corpus-based topic-sensitive language models of highly inflected languages for large vocabulary speech recognition.[1] We concentrate on feature extraction process devoted to languages where words are formed by many different inflectional affixatations. In our approach all words with the same meaning but different grammatical form are collected in one cluster automatically by using fuzzy comparison function. Using topic classifier sub-corpus of a large collection of training text is selected. Language models are built by interpolation of topic specific models and general model. Results of experiments on English and Slovenian corpus are reported.

1. INTRODUCTION

N-gram language models have proved to be surprisingly powerful [1] and easy to implement. Trigram models are most commonly used. Unfortunately, they do possess several drawbacks. N-gram techniques are unable to model long range dependencies. They lack any ability to exploit the linguistic nuances between domains.

If we could effectively identify the domain of discourse, a model appropriate for the current domain could be used. In our experiments we were looking for methods of topic adaptation in unrestricted domains. We do not assume that a document belongs to only one predefined topic. Every test sample is seen as a combination of several elemental topics.

2. TOPIC ADAPTATION

By using all training text in language model probabilities calculation, we get the model of general language.

The goal of the adaptation is to lower the language model perplexity by providing a higher probability of words and word-sequences which are characteristic of the domain of discourse.

[1]This work was funded by the Ministry of Science and Technology, Slovenia, under the contract number 3411-98-22-0854.

Initially, we made an experiment where we measured the perplexity of test samples, for which the topics were known in advance. We built three types of models:

- general language model(G). It was built by using all available training text.
- topic models(T). They were built by using only topic-specific training text.
- combined models(C). We wanted to design models which are able to predict general language and deliver a degree of topic specialisation. To meet this requirements combined models were created by interpolating topic models and a general model:

$$P_C(w) = \lambda P_T(w) + (1-\lambda)P_G(w). \qquad (1)$$

The results have shown that the perplexities given solely by the topic models are the worst. The volume of text available to create a topic model is only a fraction of that available for general model. This leads to the data sparseness problem. The combined model gave the best results for all test samples.

It is unlikely that sufficient training material will exist to create a good model of a predefined topic. In most cases we have only a sample from the target environment. With the growing availability of textual data in electronic form large topically diverse corpora are constructed. We want to develop an adaptation scheme which will be able to extract target-topic-similar parts of the whole collection and treat them as more representative [3].

The topic adaptation scheme usually consist of the following three steps [7]:

- corpus organisation. Stories that share similar topics are gathered together into a set of clusters.
- topic classification. A classifier is used to find the clusters that are most similar in topic to the test sample.
- language model building. Language models are built based on data found to be most similar to the test sample. The models are interpolated at the word level.

3. REPRESENTATION

Given a corpus with keywords assigned to each story, topic clusters are simply created by defining each keyword as a label for a cluster. Unfortunately for minority languages such corpora is often not available. Automated generation of clusters of documents based on some similarity measure need to be used.

First we have to find the suitable representation. Documents and clusters are represented as a set of features $d_i = [f_1, f_2, ..., f_m]$. In most applications words are used as features. In contrast to n-gram modelling, word order is ignored, which is of course in line with the semantic nature of the approach. It has been argued that maximum performance is often not achieved by using all available features, but using a good subset of those only [9] [4].

Having features which do not help discriminate between topics adds noise. Such words are for example non content words. In the first step we eliminate them. We used the stop-word list[2] for English language. In experiments on Slovenian language we simply removed most frequently occurring words.

3.1. Word Clustering

In this paper we would like to show that it makes sense to group features into clusters, at least for languages with rich inflectional morphology. We want to group all words with the same meaning (but different grammatical form) in one cluster and represent them as one feature. We propose a novel approach for feature extraction based on soft comparison of words [10].

To avoid the use of an additional knowledge source (lexicon) we define a membership function ($\mu_{\tilde{c}}$), which associates to each word ($w \in \vartheta$) a number representing the grade of membership of this word in a cluster of words with the same meaning ($c \in \mathcal{C}$). $\mathcal{C}$ denotes a set of clusters. Cluster c is defined as fuzzy set $\tilde{c}$:

$$\tilde{c} = \{ (w, \mu_{\tilde{c}}) \mid w \in \vartheta \}. \tag{2}$$

Each cluster defines its own fuzzy set. We want to collect all inflection forms of a lemma in one cluster automatically.

The words were compared by using a fuzzy comparison function (μ_{ϑ}). Each word w defines its own fuzzy set $\tilde{w}$:

$$\tilde{w} = \{ (w, \mu_{\tilde{w}}) \mid w \in \vartheta \}. \tag{3}$$

The function sees the word as a sequence of characters. It returns value 1 if words are the same and 0 for extremely different words. In other cases it returns the value between 0 and 1. The comparison function is created by using fuzzy rules, which provide a natural way of dealing with partial matching. The rules are expressed as fuzzy implications. The implications use linguistic variables to express similarity (for example: not very similar, quite similar). We present two examples. a denotes the word with n characters and b denotes the word with m characters. The fuzzy implication

characters of words are different
$$\Rightarrow \textit{words are not very similar} \tag{4}$$

is transformed into

$$p_1(i,a,b) = \begin{cases} 0 & \exists j : a(i) = b(j) \\ 1 & \text{otherwise} \end{cases}$$

$$e_1(a,b) = \sum_{i=1}^{n} \frac{p_1(i,a,b)}{n+m} + \sum_{i=1}^{m} \frac{p_1(i,b,a)}{n+m}. \tag{5}$$

The fuzzy implication

two character sequences of words are the same
$$\Rightarrow \textit{words are quite similar} \tag{6}$$

is transformed into the predicate

$$p_2(i,a,b) = \begin{cases} 1 & \exists j : a(i) = b(j) \wedge \\ & a(i+1) = b(j+1) \\ 0 & \text{otherwise} \end{cases}$$

$$e_2(a,b) = \sum_{i=1}^{n} \frac{p_2(i,a,b)}{n+m-2} + \sum_{i=1}^{m} \frac{p_2(i,b,a)}{n+m-2}. \tag{7}$$

The predicates are scaled by linguistic variables. Their values were empirically chosen. $e_{Similar}$ denotes the set of p_i which describe the similarity and $e_{NotSimilar}$ denotes the set of p_i which describe the distinction. The final value of comparison function is computed using scaling:

$$\mu_{\tilde{a}}(b) = \frac{\max(e_{Similar}(a,b))}{\max(e_{Similar}(a,b)) + \max(e_{NotSimilar}(a,b))} \tag{8}$$

It's easy to proof that

$$\mu_{\tilde{a}}(b) = \mu_{\tilde{b}}(a) \tag{9}$$

The similarity function was adapted to English and Slovenian language by adding language dependent crisp implications. For English language the rules are taken from the suffix stripping set of rules provided by Porter [6]. For example, the word endings like -ed, -able, -ing ... are treated as suffixes of words with the same meaning.

The simplest example is:

words differ in the suffix -s $\Rightarrow$ *words are the same* (10)

Similarity function adapted to the highly inflected Slovenian language has a automatically generated suffix list, based on reversed, alphabetically sorted list of words from the training corpus [5].

Having similarity values for word pairs, we create a cluster hierarchy by using a modified single link agglomerative clustering [8]. Similarity values of the words can be represented as a weighted, undirected graph where nodes of

[2]The list was produced by the Information Retrieval Laboratory at the University of Massachusetts at Amherst

the graph represent the words and the weight of an edge represents the similarity of words connected by the edge. To save space, we keep only the edges with weights greater then a prespecified threshold. The result of the single link hierarchy are locally coherent clusters. To avoid a chaining effect (and consequently elongated clusters) we modify the merging criterion. A word is added to the cluster if its average similarity with all words in cluster is the largest among all the words not yet clustered. The similarity between cluster c_i and word w_j is computed as:

$$similarity\ (c_i, w_j) = \frac{1}{M_i} \sum_{w_i \in c_i} \mu_{\vec{w}_i}(w_j). \qquad (11)$$

M_i denotes the number of words currently in cluster. Clusters are made one at the time. We start building a new cluster as soon as the largest similarity value does not exceed a prespecified threshold. Each cluster defines one feature. The number of clusters represents a feature vector length.

The Figure 1 shows an example of a cluster. Weights are computed using language independent rules. The threshold is set to 0.4. The word *bilingual* is selected as the first word of a cluster. Words are added to the cluster one after another as denoted by the number associated to each node. The word *multilingual* is not added to the cluster because the similarity does not exceed a threshold. This can be changed by adding crisp implication

$$\begin{aligned} &words\ differ\ in\ the\ prefix \in \quad \{bi-, multi-, ...\} \\ &\Rightarrow\ words\ are\ the\ same. \end{aligned} \qquad (12)$$

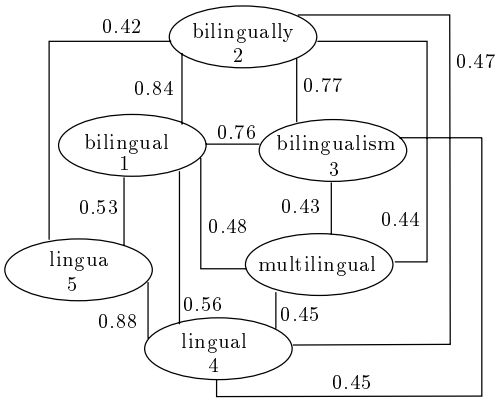

Figure 1: The example of a cluster building.

4. TOPIC DETECTION

Once we have training documents and topic clusters represented as feature vectors, we use topic detection to determine the similarity between two feature vectors. Topic detection is performed by the use of naive Bayes classifier [2, 7]. It is used to determine the similarity between two documents or clusters in document tree building for Slovenian corpus and for the topic detection of both English and Slovenian test sample. Document tree for Slovenian corpus was built by the use of agglomerative clustering [8].

5. MODEL INTERPOLATION

In addition to the three types of models (G, T, C) described in Section 2 a model named N is built by the interpolation formula:

$$P_N(w) = \lambda_1 P_T(w) + \lambda_2 P_{T_{10}}(w) + \lambda_3 P_G(w). \qquad (13)$$

Interpolation weights satisfy $\lambda_1 + \lambda_2 + \lambda_3 = 1$. The probabilities of model T are computed based on topic cluster selected by the classifier as most similar. The probabilities of model T_{10} are computed based on top 10 clusters selected by the classifier.

6. EXPERIMENTS

In our experiments we were using the broadcast news corpus (1996 CSR Hub-4 Language Model) for English language and newspaper news corpus (Večer) for Slovenian language due to their semantic richness.

The English broadcast news corpus was organised into topic-specific clusters of documents based on manually-assigned keywords. We were experimenting with topic clusters that have at least 300 articles. 244 clusters satisfy this constrain. Language model adaptation was performed on 20 randomly chosen topics. 80 % of each topic cluster text was used for language model training, 10 % for interpolation parameter estimation and 10 % was used as test sample.

All words from the corpus were used for feature extraction. Words from stop-word list were removed. Using language independent word clustering feature vector size was reduced from 170.00 to 36.000. A sample of clusters is shown in Table 1.

aadmirable admirable admirably admira admirally admire admired admirer admires admir admirers admiring admiration
bbecause becau becaue beca bec
chinasports sports sport sporto sporty sported sporter sportin sporting sportscar sportsman sportsmen sportcoat sportless
cilnton clnton cinton tonton
conferenced conferences conferencing teleconference teleconferenced teleconferences teleconferencing videoconferences

Table 1: Sample of English clusters.

For each test sample we want to model all topics were

ranked by the similarity value. If we used language dependent rules in features extraction process the top 10 topics didn't change. Four types of language models were built: general model (G), topic model (T), combined model(C) and more complex combined model(N). All language models are trigram models with the vocabulary of 64.000 most frequent words. Results of 5 topics (A - AUTOMOBILES; B - MIDDLE EAST; C - CLINTON, BILL; D - HOLIDAYS; E - SIMPSON, O. J.) are shown in Table 2. Averaging over all 20 experimental topics the perplexity of adapted language models was reduced by 15%.

Unfortunately, the Slovenian newspaper corpus is not yet annotated with keywords. Clustering was done automatically. Documents were merged iteratively until we have got less then 100 clusters.

Feature extraction was performed in the same way. Using language independent word clustering feature vector size was reduced from 140.00 to 21.000. A sample of clusters is shown in Table 3. Words in bold are from semantic point of view not correct clustered. Adding language dependent rules feature vector size was reduced to 18.000.

Three test samples were manually created (X - SPORT; Y - WEATHER FORECAST; Z - POLITICS). All of them consist of 5 documents similar in topic. Language models were built in the same way as in the previous experiment with English corpus. By using the language independent feature extraction, we have got a 14% perplexity reduction. By using to Slovenian language adapted feature extraction, the perplexity reduction was up to 30% (see Table 4).

7. CONCLUSION

In our experiments we have shown that topic adaptation does result in a decrease in perplexity. To train a language model it does not make sense to use only a small portion of topic specialised text. Experiments have shown that topic adaptation is possible even if corpus with keywords assigned to it is not available. In this case features extraction plays an important role, in particular for languages with rich morphology.

8. REFERENCES

1. F. Jelinek. *Statistical Methods for Speech Recognition.* MIT Press, 1997.

topic	PP_G	PP_T	PP_C	PP_N
A	58	247	55	53
B	55	145	27	27
C	48	90	46	41
D	62	331	49	48
E	45	59	25	23

Table 2: The sample of English test set perplexities.

afer afera aferah afere aferi afero aferami **fer**
bancna bancne bancnem bancni bancno bancnic bancnih bancnik bancnim bancnika bancniki bancnimi bancnikom bancnikov bancniski bancnega bancnistva bancnistvo bancnistvu
cestnemu mestnemu mestnem mestne mestnega mestna mestni mestno
nihanje nihanj nihanja nihanju **ihan**
dobojevati izbojevati izbojeval

Table 3: Sample of Slovenian clusters.

language independent feature extraction				
topic	PP_G	PP_T	PP_C	PP_N
X	230	621	199	197
Y	154	201	153	141
Z	220	598	210	215

language dependent feature extraction			
topic	PP_T	PP_C	PP_N
X	455	183	161
Y	198	150	115
Z	598	201	189

Table 4: The sample of Slovenian test set perplexities.

2. T. Joachims. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. Technical report, Carnegie Mellon University, Pittsburgh, USA, march 1996.
3. D. Klakow. Selecting Articles from the Language Model Training Corpus. In *Proc. ICASSP*, 2000.
4. D. Mladenić and M. Grobelnik. Feature Subset Selection in Text-learning. In *Proc. ECML*, 1998.
5. M. Popovič and P. Willett. Processing of Documents and Queries in a Slovene Language Free Text Retrieval System. *Literary and Linguistic Computing*, 1990.
6. M. F. Porter. An Algorithm for Suffix Stripping. *Program*, 1980.
7. K. Seymore and R. Rosenfeld. Using Story Topics for Language Model Adaptation. In *Proc. Eurospeech*, 1997.
8. E. M. Voorhees. Implementing Agglomerative Hierarchic Clustering Algorithms for Use in Document Retrieval. Technical report, Cornell University, Ithaca, New York, july 1986.
9. Y. Yang and J. O. Pederson. A Comparative Study on Feature Selection in Text Categorization. In *Proc. ICML*, 1997.
10. H. J. Zimmermann. *Fuzzy Set Theory - and Its Applications.* Kluwer - Nijhoff., 1986.