Topic-sensitive Language Modelling

Mirjam Sepesy Maučec and Zdravko Kačič

University of Maribor, Smetanova 17, SI-2000 Maribor, Slovenia

Abstract. The paper proposes a new framework to construct topic-sensitive language models for large vocabulary speech recognition. Identifying a domain of discourse, a model appropriate for the current domain can be built. In our experiments the target domain was represented with a piece of text. By using appropriate features, sub-corpus of a large collection of training text was extracted. Our feature selection process was especially suited to languages where words are formed by many different inflectional affixatation. All words with the same meaning (but different grammatical form) were collected in one cluster and represented as one feature. We used the heuristic word weighting classifier TFIDF (term frequency / inverse document frequency) to further shrink the feature vector. Final language model was built by interpolation of topic specific models and a general model. Experiments have been done by using English and Slovenian corpus.

1 Introduction

In this article we address the problem of constructing language models for large vocabulary speech recognition. N-gram language models have proved to be surprisingly powerful [1] and easy to implement. Trigram models are most commonly used. Through the growing availability of textual data in electronic form, n-gram models have sought some improvements in reducing the problem of sparseness. 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 this case, language models adapted to particular task need to be built.

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 of a number of disjoint topic sets. Every document, story or conversation is seen as a unique and hitherto unseen 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 [2]. Being able to cluster training documents into topic-specific clusters, topical models can be built.

Initially, we made an experiment where we measured the perplexity of test samples, for which the topics were known in advance.

10 topics (A, B, ..., J) were randomly selected from the broadcast news corpus (1996 Hub4 experiment). The stories in the corpus are annotated with keywords. By using them, clusters of stories with the same keyword were created. For all the 10 selected topics, single-topic language models were built by using merely the text in their cluster. Between all the single-topic models, the perplexity of the correct model was always the smallest.

We wanted to design models which:

- are able to predict general language and
- deliver a degree of topic specialisation.

Combined language models were created by interpolating topic models (TOP) and a general model (GEN):

$$P_{COM}(w) = \lambda P_{TOP}(w) + (1 - \lambda) P_{GEN}(w). \tag{1}$$

P_{TOP} denotes the probability given by the topic model, P_{GEN} the probability given by the general model and P_{COM} the probability given by a combined model.

Table 1. Perplexities of topic models, the general model and combined models computed on test samples of known topic.

	test sample									
model	A	B	C	D	E	F	G	H	I	J
GEN	157	58	281	63	210	142	324	143	47	222
TOP	247	146	241	91	205	174	263	145	148	229
COM	80	50	125	54	99	80	130	81	44	100

The results (see Table 1) show 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 topic test samples.

3 Topic Representation

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. From the large amount of text we want to extract documents, which are semantically most similar.

Documents and clusters are represented as a set of words, named features $d_i = [w_1, w_2, ..., w_m]$. 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. We have done experiments to show that it makes sense to group features into clusters. We want to group all words with the same meaning (but different grammatical form) in one cluster and represent them as one feature.

3.1 Fuzzy Clustering

We propose a novel approach for feature selection based on soft comparison of words. The usual way of learning on text defines a feature for each word that occurred in the training documents. This can result in some hundred of thousands of features. Having no additional knowledge source (lexicon) for this task, we define a membership function ($\mu_{\mathcal{C}}$), 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 $\tilde{c}$:

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

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 $\tilde{w}$:

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

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, which use linguistic variables to express similarity (similar, very similar, not very similar, quite similar, ...). We have made experiments with three different similarity functions. The first was constructed by using language independent rules. The rule compares parts of words (sequences of one, two, three ... characters) and implies the degree of similarity. For example:

$$\textit{tree character sequences of words are the same} \Rightarrow \textit{words are quite similar} \tag{4}$$

The second similarity function was adapted to English language. For example, the parts of words like co-, ex-, non-, pre-, re- ... were treated as prefixes and parts like -ed, -able, -ing ... as suffixes of words with the same meaning. For example:

$$\textit{words differ in the suffix -s} \Rightarrow \textit{words are the same} \tag{5}$$

This kind of rules are expressed as crisp implications.

The third similarity function was adapted to the highly inflected Slovenian language and has the greatest number of rules.

Having similarity values for word pairs, we created a cluster hierarchy by using single link agglomerative clustering. Similarity values of the words can be represented as a weighted, undirected graph where nodes of the graph represent the words and the weight of an edge represents the similarity of words connected by the edge. To save space, we kept 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 modified the merging criterion. Clusters were made one at the time. A word was added to the cluster if its average similarity with all words in cluster was the largest among all the words not yet clustered. We started building a new cluster as soon as the largest similarity value did not exceed a prespecified threshold. This approach has still a running time of $O(N^2)$ and space requirement of $O(N)$. As an example, there are some randomly selected clusters of English language presented (see Table 2).

Table 2. Example of English clusters.

Cluster No.	words in cluster
21	aadmirable admirable admirably admira admirally admire admired admirer admires admir admirers admiring admiration
3263	bbecause becau becaue beca bec
6457	chinasports sports sport sporto sporty sported sporter sportin sporting sportscar sportsman sportsmen sportcoat sportless
6645	cilnton clnton cinton tonton
7269	conferenced conferences conferencing teleconference teleconferenced teleconferences teleconferencing videoconferences
17199	iraq iraqi iraqis iraqs iraqian iraqui

3.2 Feature Vector Shrinking

Each cluster defines one feature. The number of clusters represents a feature vector length. We have further shrunk the vectors by a feature scoring measure TFIDF (term frequency / inverse document frequency), defined as:

$$TFIDF(w_i, d) = TF(w_i, d) \cdot (\log(|D|) - \log(DF(w_i)) + 1). \qquad (6)$$

$TF(w_i, d)$ is the number of times word w_i occurs in topic corpora d. $|D|$ denotes the total number of topic corpora. $DF(w_i)$ is the number of topic corpora in which the word w_i occurs at least once.

4 Topic Classification

Once we have the topic corpora ($\overrightarrow{d}$) and test documents ($\overrightarrow{t}$) represented as vectors of features, we used topic detection to determine the most similar topic clusters to a new piece of text. We used the TFIDF classifier:

$$H_{TFIDF} = arg \max_{d \in D} \frac{\overrightarrow{t} \cdot \overrightarrow{d}}{\| \overrightarrow{t} \| \cdot \| \overrightarrow{d} \|}. \tag{7}$$

5 Language Model Interpolation

After selecting k most similar topics individual trigram language models for each topic were built and interpolated together with the general model to produce a new language score [3].

$$P_{NEW}(w) = \sum_{j=1}^{k} \lambda_k P_{TOP_k}(w) + (1 - \sum_{j=1}^{k} \lambda_k) P_{GEN}(w). \tag{8}$$

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.

All language models were trigram models with the vocabulary of 64.000 most frequent words.

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 satisfied this constrain. For a new piece of text we want to model, 10 most similar topics were extracted. Ranking all the topics by similarity value from 1 to 244, the correct topic cluster was always between the top 10 clusters (see Table 3). Language models built from each of the top 10 topic clusters were interpolated with a general language model. By using the language independent similarity function we achieved a 10% perplexity reduction. By using to English language adapted similarity function, the perplexity reduction was slightly better (14%) (see Table 3). PP NEW denotes the perplexity of a new language model and topics selected by adapted similarity function.

Unfortunately, the Slovenian newspaper corpus is not yet annotated with keywords. Clustering was done automatically. At the beginning each document defined its own cluster. We merged clusters iteratively by using the TFIDF classifier until we have got less then 100 of them. Three test samples were manually created. All of them consist of 5 documents similar in topic. For every sample, 10 most similar clusters were extracted. Language models were built in the same

Table 3. Rank of correct topic, perplexity of a new model and improvement of a new model in comparison with the combined model.

	test sample									
	A	B	C	D	E	F	G	H	I	J
rank of correct topic	1	8	6	9	8	4	3	8	7	3
PP NEW	69	52	110	53	91	71	112	79	45	89
Improvement [%]	14	-4	12	2	8	11	14	2	-2	11

way as in the previous experiment with English corpus. By using the language independent similarity function, we have got a 15% perplexity reduction. By using to Slovenian language adapted similarity function, the perplexity reduction was up to 30% (see Table 4).

Table 4. Perplexity of a general and new model and improvement of the new model in comparison with the general model.

	test sample		
	X	Y	Z
PP GEN	230	154	220
PP NEW	161	115	189
Improvement [%]	30	25	14

7 Conclusion

In our experiments we have shown that topic adaptation does result in decrease in perplexity. Adapted models can exploit the localisation of information within a topic domain. To train a language model, it does not make sense to use only a small portion of the topic specialised text. Large volumes of unrestricted text should be used to keep the characteristics of language in general.

References

1. Jelinek F.: Statistical Methods for Speech Recognition. MIT Press (1997)
2. Donnelly P. G., Smith F. J., Sicilia E., Ming J.: Language Modelling with Hierarchical Domains. Proc. Eurospeech (1999)
3. Seymore K., Rosenfeld R.: Using Story Topics for Language Model Adaptation. Proc. Eurospeech (1997)