Multi Domain Language Model Adaptation using Explicit Semantic Analysis

Kevin Kilgour[1,2]*, Florian Kraft*[1] *Sebastian Stüker*[1,2] *and Alex Waibel*[1]

[1]Institut für Anthropomatik
[2]Research Group 3-01 'Multilingual Speech Recognition'
Karlsruhe Institute of Technology
Karlsruhe, Germany
{ytitov|kevin.kilgour|stueker|waibel}@ira.uka.de

Abstract

This paper presents an adaptive multi domain language model built from large sources of pre existing human created structured data. The sources' structure is exploited to create a large array of ngram language models which are dynamically interpolated at decoding time to produce a context dependent language model that continuously adapts itself to the current domain. Because the use of human annotators is expensive and impractical we explore existing sources of human created structured data and how to extract our desired data from them.

The language model is evaluated on its performance with a speech recognition system used to decode the Quaero 2009 evaluation data set. Compared to the baseline language model of our Quaero 2009 evaluation system our proposed adaptive language model reduces the WER of the speech recognition system by 0.5% absolute with some shows showing reductions of up to 14.4%.

Index Terms: Speech Recognition, Language Model Adaptation, Explicit Semantic Analysis

1. Introduction

Most state of the art language models used in speech recognition and machine translation are statistical language models trained from a large amount of language data. The better the data represents the type of language the model will encounter, the better the language model will be. This means that having more data does not necessarily lead to a better language model. The data has to be relevant to the domain of the language model.

It can easily be seen that language models built for a particular domain will perform better on that domain than a general purpose language model. A language model used to transcribe business meetings, for example, will pretty much never have to deal with the sequence of words *"high altitude cerebral edema"* whereas a general purpose language model that could also be used in hospitals and by rock climbers would have to deal with this and many other equally diverse word sequences. As general purpose language models do not perform as well as domain language models in their domains they could be improved by adapting themselves to the current domain.

There are many online sources containing domain specific texts that could be used as language model training data for that particular domain. The problem we face is not so much an absence of text data but rather the absence of domain categorized text data. After a brief overview of other adaptive language models in section 2, section 3 of this paper describes the Open Directory Project (ODP) out of which large amounts of categorized texts can be extracted and goes on to to explain how Explicit Semantic Analysis (ESA) can be applied to them. In section 4 we present our adaptive language model which is then compared to a baseline ngram language model in section 5 with conclusions in section 6.

2. Related Work

The first adaptive language model was a cache-based language model designed by Kuhn, De Mori, McGill [1]. A cache-based language model simply increases the probability of a word whenever it is observed, assuming that words that appear once are more likely to appear again. To cope with a change of speaker or topic the probability increase from observing a word is not permanent. A cache-based language model is composed of two weighted probabilities, the cache part and the original n-gram part. To more accurately model the observation that, *"the more recent an occurrence, the higher the probability of a re-occurrence"*, Clarkson and Robinson added a decaying factor to the cache function [2].

A cache based language model that observes the word *Obama* will correctly increase the probability of detecting *Obama* again but it will not change the probability of detecting *Barack*[1]. To take advantage of these word correlations Lau and Rosenberg built a trigger language model that not only increases the probability of a word when it is seen but also increases the probabilities of those words that are highly correlated with it [3]. Their language model had a perplexity that was 13% lower than that of a comparative tri-gram language model.

Mixture language models ([4]) are another extension of cache language models that consist of *k* n-gram language models pre-trained on different text catagories like *scientific writing* or *newspaper text* which are then interpolated at run-time. Using maximum-likelihood

[1] Actually, because the probability density remains the same, by increasing the probability of *Obama* the probabilities of all other words including *Barack* are reduced by a tiny amount

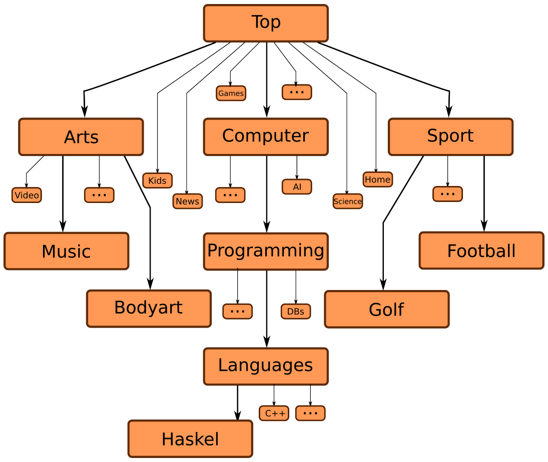

Figure 1: Schematic of the Open Directory Project. Each node contains links to multiple website and represents a concept; the associated concept language model is construed from the its linked to websites as well as all the websites linked to by its daughter concepts.

estimation the interpolation weights λ are calculated to maximise the likelihood of the word sequence in the cache.

$$P_{mixture}(w_i|h) = \sum_{j=1}^{k} \lambda_j\left(h\right) P_j(w_i|h) \tag{1}$$

Using $k = 15$ this approach showed a 10% reduction in perplexity compared to a standard bi-gram language model.

Another extension to the ngram language model, called *factored language model* [5], where words are represented by tuples of features (classes, stem, morphology etc.) and the probabilities of tuple sequences modeled, was shown to reduce the WER on the LDC CallHome corpus (Arabic) by 1.8% absolute.

Using Latent Dirichlet Allocation (LDA), a variant of Latent Semantic Analysis, Tam and Schultz built an adaptive language model with 200 latent topics [6] and showed a 0.4% decrease in CER on the Manderin RT04-eval set. Unfortunately the use of LDA adapted language models have so far not been shown to improve the WER on English recognition tasks.

3. Explicit Semantic Analysis

In the better known Latent Semantic Analysis [7] a document or sequence of words (from a speech recogniser for example) is represented as a vector in a latent concept space. The concepts are generated algorithmically and are not necessarily comprehendable to people. In contrast Explicit Semantic Analysis (ESA) developed by Gabrilovich and Markovitch [8] represents the word sequence as a vector in a concept space where the concepts have already been defined by humans. Most commonly the articles in the online encyclopedia Wikipedia are used as the explicit concepts. The word sequence is compared to each of the concepts and with a tfidf measure (see section 3.2) the importance of the concept to the word sequence is determined.

3.1. Open Directory Project (ODP)

As we require a large amount of text to be associated with each concept, we chose the categories in the open directory project (ODP)[2] to be our explicit concepts. The ODP is a large hierarchically sorted directory of websites which is edited and maintained by human volunteers. It contains links to over 4 million websites sorted into over 500,000 concepts. As can be seen in figure 1 its hierarchical tree-like structure allows us to associate the text in the websites which are linked to by a concept, not only with their corresponding concepts, but also with all their ancestor concepts. The k concepts C with the most associated[3] text are chosen to be the dimensions in our concept space $\mathcal{C}_k$ giving us a mapping function:

$$\tau : \mathcal{D} \rightarrow \mathcal{C}_k \subset \mathbb{R}^k \tag{2}$$

$$\tau\left(w_i\right) = \left\langle \upsilon\left(w_i, c_1\right), \upsilon\left(w_i, c_2\right), \ldots, \upsilon\left(w_i, c_k\right)\right\rangle \tag{3}$$

Because both the concepts c_j and the word sequence w_i can be considered documents in the text classification sense, we can use text classification methods to find the similarity υ between a word sequence w_i and a concept c_j ($w_i, c_j \in \mathcal{D}$, the set of all possible documents). Documents (word sequences or concepts) have lots of properties that can be useful in deciding how similar they are to one another. By far the most important property is its body of text form which term counts (in our case word counts) are extracted and a feature vector is built.

[2] http://www.dmoz.org/

[3] Concepts containing over 10% of the total ODP data are disregarded, because we considered them to be too general.

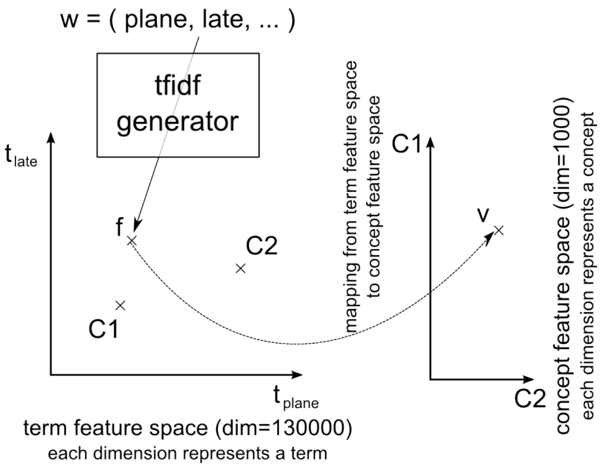
w = (plane, late, ...)
tfidf generator
t_{late}
f
C2
C1
t_{plane}
term feature space (dim=130000)
each dimension represents a term
mapping from term feature space to concept feature space
C1
v
C2
concept feature space (dim=1000)
each dimension represents a concept

Figure 2: Explicit Semantic Analysis performed on a short sequence of words. A sequence of words is first of all converted to a sparse term feature vector using tfidfs and then mapped to a vector in the concept feature space by comparing it to the term feature vectors of each concept.

3.2. *Term-Frequency Inverse Document Frequency (TFIDF) Metric*

A standard method of generating a feature vector $\overrightarrow{f_j}$ from a document c_j is to first extract a set of n terms from the sum of the text of all the documents and then weight them according to their occurrence in c_j. For each c_j we have a

$$\overrightarrow{f_j} = (f_{1,j}, f_{2,j}, ..., f_{n,j}) \tag{4}$$

where $f_{i,j}$ denotes the weight of term i in document j.

Once a set of terms $|\mathcal{T}| = n$ has been decided upon a TFIDF function can be used to generate a document feature vector $\overrightarrow{f_j}$. In our case the number of terms is limited by the vocabulary of our speech recogniser. The function calculates each component of $\overrightarrow{f_j}$ from its term frequency $\text{TF}_{i,j}$ in c_j and the inverse document frequency IDF_i of $t_i \in \mathcal{T}$. The term frequency component measures how often a word occurs in a document. Let $\#(t_i, c_j)$ be the number of times t_i appears in c_j.

$$\text{TF}_{i,j} = \log \#(t_i, c_j) \tag{5}$$

The inverse document frequency measures how discriminative a word is. Words that appear in few documents have a high inverse document frequency and words in a lot of documents have a low inverse document frequency [9].

$$\text{IDF}_{i,j} = \log\left(\frac{|C|}{\#(C, t_i)}\right) \tag{6}$$

$\#(C, t_i)$ is the number of documents containing t_i. The logarithm of the quotient is used to blunt the effect of extremely rare words, which might only appear in one or two documents. Putting these together gives us.

$$f_{i,j} = \text{TFIDF}(t_i, c_j) = \log \#(t_i, c_j) \cdot \log\left(\frac{|C|}{\#(C, t_i)}\right) \tag{7}$$

As is, the TFIDF function does not take into account the length of a document. A term appearing once in a short document is more relevant than if it were to appear in a longer one.

One way to solve this is to normalise the vector generated by the TFIDF function.

$$f_{i,j} = \frac{\text{TFIDF}(t_i, c_j)}{\sqrt{\sum_{h=1}^{|\mathcal{T}|} \text{TFIDF}(t_h, c_j)^2}} \tag{8}$$

The same feature extraction method is applied to the new document or word sequence w that is to be mapped to the concept space.

$$f_{i,w} = \log \#(t_i, w) \cdot \log\left(\frac{|C|}{\#(C, t_i)}\right) \tag{9}$$

$$\phi_{\mathcal{T}}(w) = \overrightarrow{f_w} = (f_{1,w}, f_{2,w}, ..., f_{n,w}) \tag{10}$$

Because the selection of our terms $\mathcal{T}$ will remain constant we will henceforth refer to $\phi_{\mathcal{T}}$ simply as ϕ.

	Single Pass	Two Pass
Utt History	Perplexity	Perplexity
0	**147.30**	**118.54**
1	124.65	120.09
2	123.96	120.70
3	124.06	121.94
4	123.99	122.08

Table 1: *Perplexity measured on a subset of the 2006 TC-STAR development data using different approaches for setting h_{adapt}. Single Pass: h_{adapt} only includes the text from the last 0-4 utterances, with 0 being the unadapted baseline. Two Pass: as single pass but including the hypothesis generated in a first unadapted decoding pass.*

3.3. Cosine Similarity Metric

TFIDF vectors are built to be able to compare documents with each other; this requires a similarity metric. The cosine similarity metric is a simple and fast metric to compute [10]. It's application in information retrieval has been extensively studied ([11] and [12]) and it has been found to be one best similarity metrics when comparing sparse document vectors [13]. The cosine similarity measurres the angle τ between the two vectors f_1 and f_2 that are to be compared.

$$\mathrm{cossim}(f_1, f_2) = \cos(\varphi) = \frac{f_1 \cdot f_2}{|f_1||f_2|} \tag{11}$$

Since the TFIDF vectors are often already normalised ($|f_1| = 1$ and $|f_2| = 1$) the denominator part of this definition can be ignored. Also most TFIDF vectors are sparse, leading to very few non-zero terms in the numerator of the definition. This allows the cosine similarity metric to be calculated very fast, making it an ideal function for υ which we can now define as.

$$\upsilon\left(w_i, c_j\right) = \phi(w_i) \cdot \phi(c_j) \tag{12}$$

4. Language Model Adaptation

Using ESA we wish to identify the domain in which the language model is working and interpolate the original base language model P_{base} with a language model built specifically for the detected domain P_d. Let $h = w_1 w_2 ... w_{(i-1)}$ be the current word history.

$$P_{adapt}(w_i|h) = \mu P_{base}(w_i|h) + (1 - \mu) P_d(w_i|h) \tag{13}$$

In practice a text classifier might not be able to accurately identify the exact concept of the text spoken so far, or it might by possible to classify it as being associated with two or more concepts. Inserting h into τ (see equation 3) results in a k dimensional vector with $\tau\left(h\right)_j$ representing the similarity between h and concepts c_i. For computational reasons it is necessary to define:

$$\widetilde{\tau\left(h\right)_j} = \begin{cases} \tau\left(h\right)_j & \text{if } \tau\left(h\right)_j \text{ is one of the } n \text{ largest} \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

Assume without loss of generality that $c_1 ... c_n$ are the concepts for which $\widetilde{\tau\left(h\right)_j} \neq 0$. We then define their similarity weights $\lambda_1 ... \lambda_n$ as

$$\lambda_j\left(h\right) = \frac{\widetilde{\tau\left(h\right)_j}}{|\widetilde{\tau\left(h\right)}|}. \tag{15}$$

For a given word history h we now have its n most similar concepts $c_1 ... c_n$ and their normalised similarity weights $\lambda_1 ... \lambda_n$. Because using a maximum-likelihood estimation to determin the best interpolation weights turned out to be to slow we just used the normalised similarity weights.

$$P_d(w_i|h) = \sum_{j=1}^{n} \lambda_j\left(h\right) P_{c_j}(w_i|h) \tag{16}$$

The domain dependant LM is hereby constructed on the fly from a number of concept LMs.

These concept language models are pre-built from the text associated with the concepts using the SRI Language Modelling Toolkit [14]. Exploiting the hierarchical structure of the ODP allows us to construct large concept language models. Included in the data used to build the language model for the concept *"computer"* (see figure 1), for example, are not only the texts of the websites linked to in the ODP concept *"computer"* but also the texts of the websites linked to in ODP concepts *"AI"*, *"Programming"*, *"C++"* etc. For the following experiments we used 1000 concepts, the largest concept contained about 300 million words and the smallest concepts about 16 million words, with median lying around 30 million words per concept. On avarage each website linked to in the ODP was included in 8 concepts.

	Quaero 2009 data (WER)	
	dev set	eval set
Baseline LM	41.0%	31.5%
Base + ODP text data	41.3%	31.4%
Base + adaptive LM	40.6%	31.0%

Table 2: *Evaluation results of our adaptive language model compared to a baseline language model and the baseline language model augmented with the texts used to build the concept language models.*

5. Experimental Setup

All experiments in this work were performed with the help of the Janus Recognition Toolkit (JRTk) featuring the IBIS single pass decoder [15].

The acoustic model used in our experiments utilises 3-state sub-phonetically tied semi-continuous Hidden Markov Models composed of 16,000 quinphone models over 4,000 codebooks with a maximum of 64 Gaussians per model. The preprocessing stacks 15 frames of 15 mel scaled warped Minimum Variance Distortionless Response (wMVDR) cepstral coefficients [16].

The resulting feature vector is reduced to 42 dimensions using linear discriminant analysis. The model was trained on 140 hours of transcribed speech data composed of European Parliamentary Plenary Sessions [17], conference talks given by non-native speakers from the Translanguage English Database (TED) [18], and broadcast news recordings. The training procedure consisted of merge-and-split training on samples extracted with the help of existing forced alignments using one global semi-tied covariance (STC) transformation [19], followed by two iterations of viterbi training to compensate for wrong alignments. The models were then further improved by several iterations of minimum mutual information estimation (MMIE) training.

The base language model used is an interpolation of 4-gram language models trained on transcripts of European Parliamentary Plenary Sessions, news texts, the Gigaword Corpus, and data collected from the World Wide Web, for a general English transcription task. The tuning text used to estimate the interpolation values is taken from the Quaero 2009 training data transcripts, while a furthar 4-gram language model is built form the remaining transcripts. The tuning text is used to estimate the interpolation weights with the help of maximum-likelihood estimation. The models are built and interpolated using the SRI Language Modelling Toolkit [14]. The resulting language model is pruned to slightly more than 6x10^7 3-grams and 4-grams.

A further language model is built that also included the texts used to train the concept language models in its training data. Again 4-gram language models are constructed from of each of training texts and interpolated based on weights estimated on the aforementioned tuning text. Due to resource limitations we had to restrict the number of concept language models n used in the adaptive language model to 10.

5.1. Perplexity Test

Since recomputing the $\lambda_j\,(h)$ for each small change in h is very computationally intensive we modified (16) so that the λ_j are dependant on a word history h_{adapt} that will change less frequently.

$$P_d(w_i|h) = \sum_{j=1}^{n} \lambda_j\,(h_{adapt})\,P_{c_j}(w_i|h) \tag{17}$$

Our initial perplexity tests to determine which values of h_{adapt} would produce the best results were carried out on a small subset (144 utterances) of the 2006 TC-STAR development data, derived from recordings of the European Parliament plenary speeches. The utterances' length was on average 44.8 words with a standard deviation of 28.0. Two approaches for selecting h_{adapt} were compared. The first approach, needing only a single decoding pass, consisted of constructing h_{adapt} from the hypotheses of the previous utterances whereas in the other approach, requiring a second pass, h_{adapt} is simply set to the hypothesis generated in the first unadapted decoding pass. The results of these two approaches as well combinations thereof can be seen in table 1. In the single pass column h_{adapt} only includes the text from the last 0-4 utterances, with 0 being the unadapted baseline. The other column also includes the hypothesis of the first unadapted decoding pass. For example, in the two pass system with an utterance history of 2 the language model, when used to recognise an utterance, is adapted to the recognised text of of the unadapted first pass combined with the recognised text of the previous two utterances.

All configurations of h_{adapt} reduced the perplexity compared to the baseline of 147.30. Approach two, just the unadapted hypothesis and without including any of the previous hypotheses, performed the best reducing the baseline perplexity by almost 20% to 118.54. All further experiments were therefore performed with this setup.

5.2. Word Error Rate Evaluation

For measuring WER the more challenging Quaero 2009 development and evaluation sets were used. In total the Quaero 2009 evaluation set consists of just over 200 minutes of spoken audio by roughly 110 different speakers including some with strong accents. Except for a small amount of European Parliament plenary speeches and broadcast news most of the audio was extracted from podcasts found on the web. These contained both prepared and spontaneous speech from a large variety of domains. Our out of vocabulary rate on this evaluation set was 0.52%. The 11 hour long Quaero 2009 development set has a similar makeup with about 223 different speakers.

All three tested systems varied only in their language models. The baseline system used the original language model built for the 2009 Quaero evaluation. This language model was also used as the base language model in the adaptive system and was interpolated

show type	show name	baseLM (in %)	adaptLM (in %)	rel. gain (in %)
Lecture	17437	21.3	21.3	0
Lecture	17446	27.7	23.7	**14.4**
EPPS	20080605_0902_1100_or	11.4	11.4	0
EPPS	20080617_1530_1759_or	15.2	15.2	0
Lecture	jodcast_nightsky	27.3	24.2	**11.4**
News	bbcglobalnews	27.1	27.0	0.4
News	channel4_20080709	33.9	33.8	0.3
Discussion	creepy_crawlies	22.2	23.6	-6.3
Discussion	development_article	7.8	7.8	0
Discussion	energy	27.5	27.6	-0.4
Interview	honourcrime	30.9	32.4	-4.9
Interview	jamie_oliver_on_qtv	49.2	47.8	2.8
Discussion	naked_scientists_08_06_22	38.0	37.4	1.6
Discussion	naked_scientists_08_07_20	36.3	35.7	1.7
News	ch4ln_news	31.3	30.6	2.2
Discussion	sciencesnaps_2007	61.6	57.4	6.7
Interview	sound	55.5	50.5	9.0

Table 3: *Results of the baseline LM and the adaptive LM on the individual shows included in the Quaero 2009 evaluation set.*

with the adaptive language model using a μ of 0.8. A further system was built to ensure that it is the adaptation method that is responsible for the decrease in WER and not just the extra text data we used build the concept languages. It used a language model built from the text used to build the baseline language model as well as the texts used to build the concept language models.

As can be seen in table 2 our adaptive language model improves upon the baseline WER of 31.5%, reducing it by 0.5% absolute to 31.0%. This improvement is considerably better than the 0.1% improvement achieved by the augmented language model.

An examination of the adaptive language model's improvements on a each show in evaluation set (see table 3) indicates that, instead of a uniform improvement, some shows benefit more than others. Two shows of the type lecture showed reductions in WER of over 10% (14.4% & 11.4%). The large variance in WER chance indicates that further research has to be carried out in order to determin which parameters correlate with a change in WER, allowing us to also dynamically adjust the mixture weights of the base language model μ and the domain adapted language model $1 - \mu$.

6. Conclusion

This paper has presented an explicit semantic analysis based adaptive language model. Because we utilise the existing categorisation efforts performed by the human volunteer editors of the open directory project we are able to automatically extract large amounts of labled data without human assistance. The ODP's hierarchically structure is also exploited to increase the training data for the individual concept language models.

Through the use of text classification methods the language model is able to dynamically adapt itself to the current domain or topic. As well as being deployable in any domain it's adaptive nature also makes it robust to topic changes.

Unlike many approches at building adaptive language models the method proposed in this paper not only reduces the perplexity of a our English test set but is also able to reduce the WER. With an overall WER reduction of 0.5% absolute and some shows showing WER reductions of up to 14.4% this can be seen as an important step towards a general purpose multi domain language model that does not have to be manually tweaked for each new domain it is used in.

7. References

[1] R. Kuhn, R. De Mori, McGill University, and School of Computer Science, "A cache-based natural language model for speech recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 6, pp. 570–583, 1990.

[2] PR Clarkson and AJ Robinson, "Language model adaptation using mixtures and an exponentiallydecaying cache," in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, 1997, vol. 2.

[3] R. Lau, R. Rosenfeld, and S. Roukos, "Trigger-based language models: A maximum entropy approach," in *icassp*. IEEE, 1993, pp. 45–48.

[4] R. Kneser and V. Steinbiss, "On the dynamic adaptation of stochastic language models," in *IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS SPEECH AND SIGNAL PROCESSING*. INSTITUTE OF ELECTRICAL ENGINEERS INC (IEE), 1993, vol. 2.

[5] J.A. Bilmes and K. Kirchhoff, “Factored language models and generalized parallel backoff,” in *Proceedings of HLT/NAACL*, 2003, pp. 4–6.

[6] Y.C. Tam and T. Schultz, “Unsupervised language model adaptation using latent semantic marginals,” in *Ninth International Conference on Spoken Language Processing*. Citeseer, 2006.

[7] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman, “Indexing by latent semantic analysis,” *Journal of the American Society for Information Science*, vol. 41, no. 6, 1990.

[8] E. Gabrilovich and S. Markovitch, “Computing semantic relatedness using wikipedia-based explicit semantic analysis,” in *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2007, pp. 6–12.

[9] G. Salton and C. Buckley, “Term weighting approaches in automatic text retrieval,” 1987.

[10] G. Salton and M.J. McGill, *Introduction to modern information retrieval*, McGraw-Hill New York, 1983.

[11] J. Zobel and A. Moffat, “Exploring the similarity space,” in *ACM SIGIR Forum*. ACM, 1998, vol. 32, p. 34.

[12] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval* 1,” *Information processing & management*, vol. 24, no. 5, pp. 513–523, 1988.

[13] A. Singhal, “Modern information retrieval: A brief overview,” *IEEE Data Engineering Bulletin*, vol. 24, no. 4, pp. 35–43, 2001.

[14] A. Stolcke, “SRILM-an extensible language modeling toolkit,” in *Seventh International Conference on Spoken Language Processing*. ISCA, 2002.

[15] H. Soltau, F. Metze, C. Fügen, and A. Waibel, “A one pass-decoder based on polymorphic linguistic context assignment,” in *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU ’01)*, December 2001.

[16] M. Wölfel and J.W. McDonough, “Minimum variance distortionless response spectral estimation, review and refinements,” *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 117–126, Sept. 2005.

[17] Christian Gollan, Maximilian Bisani, Stephan Kanthak, Ralf Schlüter, and Hermann Ney, “Cross domain automatic transcription on the tc-star epps corpus,” in *Proceedings of the 2005 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP ’05)*. March 2005, IEEE.

[18] Lori F. Lamel, Florian Schiel, Adrian Fourcin, Joseph Mariani, and Hans G. Tillmann, “The translanguage english database (ted),” in *Proceedings the Third International Conference on Spoken Language Processing (ICSLP 94)*. September 1994, ISCA.

[19] M.J.F. Gales, “Semi-tied covariance matrices for hidden markov models,” Tech. Rep., Cambridge University, Engineering Department, February 1998.