# Unsupervised Vocabulary Selection for Domain-Independent Simultaneous Lecture Translation

**Paul Maergner**
Carnegie Mellon University,
Karlsruhe Institute of Technology
paul.maergner@sv.cmu.edu

**Ian Lane**
Carnegie Mellon University
lane@cs.cmu.edu

**Alex Waibel**
Carnegie Mellon University,
Karlsruhe Institute of Technology
waibel@cs.cmu.edu

## Abstract

In this work, we investigate methods to automatically adapt our simultaneous lecture translation systems to the diverse topics that occur in educational lectures. Utilizing materials that are available before the lecture begins, such as lecture slides, our proposed framework iteratively searches for related documents on the World Wide Web and generates lecture-specific models and vocabularies based on the resulting documents. In this paper, we propose a novel method for vocabulary selection, a critical aspect of simultaneous translation systems where the occurrence of out-of-vocabulary words significantly degrades intelligibility. We propose a novel approach based on feature-based ranking and evaluate the effectiveness of 21 different features and their combinations for this task. On the interACT German-English simultaneous lecture translation system our proposed approach significantly improved vocabulary coverage, reducing out-of-vocabulary rate, on average by $60\%$ and up to $84\%$, compared to a lecture-independent baseline. Furthermore, a 40k vocabulary selected using our method obtained better coverage than a lecture-independent 300k vocabulary, improving intelligibility and reducing the latency of the end-to-end system.

## 1 Introduction

Education is becoming an increasingly global activity. Lectures and research presentations can be broadcasted live across educational institutes around the world enabling students access to exceptional educational content no matter their physical location. However, although physical barriers are reduced using these technologies, language barriers remain. Lectures may be given in a language different from the student's native tongue and often the students that could benefit the most from this content may not have sufficient language skills to understand the lecture unaided. Interpreters are not a practical solution in many cases as the costs involved are prohibitively high. Recent works (Fügen, 2009; Kolss et al., 2008) have thus investigated the use of speech-translation technologies to translate lectures in real-time. The biggest downfall of these systems however is portability. These systems currently only perform well if topic-specific models trained from similar lectures are available. For each new topic, significant effort and cost is required to manually transcribe and translate similar lectures, without which the system will generally perform poorly. In this work, we propose to overcome this limitation by introducing approaches to automatically adapt speech translation systems to the diverse topics that occur in educational lectures. Utilizing materials that are available before the lecture begins, such as lecture slides, our proposed framework iteratively searches for related documents on the World Wide Web and generates lecture-specific models and vocabularies based on these documents.

One critical aspect for effective spoken language translation is vocabulary coverage. If a word is not present in the active system vocabulary then it cannot be recognized or translated and is often dropped from the system output. When the mismatch between the training data used to build a spoken language translation system and the topic of conversation is severe, vocabulary coverage is poor leading to a high number of out-of-vocabulary (OOV) words, poor translation quality and low intelligibility. For effective adaptation vocabulary coverage is a key component that prior works have often overlooked.

In (Kolss et al., 2008), a system for translating German lectures into English was introduced. They selected the system vocabulary based on word occurrence counts in both in-domain (lecture transcriptions, presentation slides and web data) and out-of-domain corpora and built lecture-independent models for speech recognition and machine translation using these corpora. (Munteanu et al., 2007) introduced an approach for language model adaptation which leveraged the documents available on the World Wide Web to aid the archiving and search of lectures. Their method collected PDF documents from the WWW based on search queries extracted from the original lecture slides. This approach improved transcription accuracy compared to a lecture-independent baseline but vocabulary adaptation was not considered thus limiting the usefulness of their approach. An approach for joint vocabulary and language model adaptation was introduced in (Yamazaki et al., 2007) in which words from the lecture slides were first added to the active system vocabulary and then language model adaptation was performed using an approach similar to that described in (Munteanu et al., 2007). A similar approach was applied for automatic subtitling of lectures for the hearing impaired in (Kawahara et al., 2008; Kawahara, 2010) with an additional step in which language model adaptation was performed independently for each slide, resulting in an adaptive language model which followed the course of the ongoing lecture. Within the MIT Spoken Lecture Processing Project (Glass et al., 2007) a lecture-specific vocabulary was extracted from manually provided supplemental text provided by the lecturer, including lecture slides, journal articles, and book chapters, which are available prior to the lecture.

Although the adaptation approaches described above are effective for language model adaptation they do not significantly improve vocabulary coverage. Even when all words that are occur in the lecture slides are added to the active vocabulary, the out-of-vocabulary rate often remains high compared to using topic specific vocabularies. In this work, we propose a novel approach to improve vocabulary coverage based on a feature-based vocabulary ranking scheme and documents collected from the WWW. Our proposed approach significantly improves vocabulary coverage compared to a lecture-independent system and further improves the effectiveness of other adaptation approaches including both language model adaptation for speech recognition (Munteanu et al., 2007) and adaptation of machine translation models based on comparable corpora (Vogel, 2003).

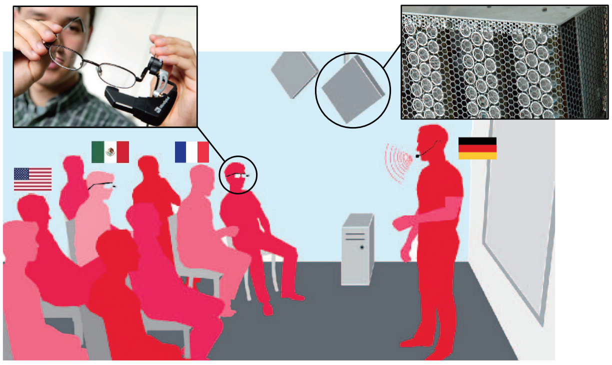

Figure 1: The interACT Lecture Translation System.

## 2 The interACT Simultaneous Lecture Translation System

The interACT Simultaneous Lecture Translation System (Fügen, 2009; Kolss et al., 2008) is a real-time lecture translation system developed at the international center for Advanced Communication Technologies (interACT) at Karlsruhe Institute of Technology (Germany) and Carnegie Mellon University (USA). This system, illustrated in Figure 1, simultaneously translates lectures in real-time from the speaker's language into multiple languages required by the audience. To minimize the distraction to the audience, our system delivers translation as either text or speech output. The translated text is displayed either on screens in the lecture room, on a website accessible on mobile devices or on heads-up displays. These technologies are especially useful for listeners who have partial knowledge of a speakers language and want to have supplemental language assistance. Spoken translation output can be listened to either via headphones or targeted audio

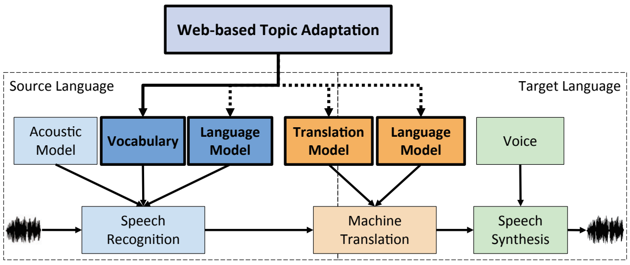

Figure 2: Components of the Lecture Translation System

speakers, which make it possible to send the translated audio only to a small group of people while the other listeners are not disturbed.

Figure 2 illustrates the three main components of our lecture translation system: Automatic *speech recognition* (ASR), *machine translation* (MT), and *speech synthesis* (Text-to-Speech, TTS). The ASR component consists of three models: An acoustic model, which models the phonetic units in the input speech, a recognition vocabulary and a source language model which models the likelihood of word sequences. The MT component consists of a translation model, which generates and scores translation alternatives in the target language, and a target language model which generates likelihoods for competing word sequences.

Input speech from the lecturer is recognized by the ASR component (Soltau et al., 2001) and the resulting output is segmented into sentence-like units which are then passed to MT. The ASR output is then translated into one or more target languages via our statistical machine translation engine STTK (Vogel et al., 2003). The translated text is either directly displayed to attendees or optionally converted into speech output using a TTS engine.

In this work, we introduce a web-based topic adaptation approach which adapts the four models indicated in Figure 2. Adaptation is performed using documents related to the lecture at hand, for example slides or lecture notes. In this paper, we focus on vocabulary selection for the speech recognition component, but our proposed approach can also be applied to adapt source and target language models and the translation model.

## 3 Web-based Vocabulary Adaptation

The vocabulary used during a lecture can be seen as a combination of two vocabularies (Glass et al., 2004; Park et al., 2005): A topic-independent lecture vocabulary, which contains vocabulary common to spontaneous speech, and a topic-dependent vocabulary. Our proposed approach for vocabulary selection uses a similar breakdown. We begin with a topic-independent lecture vocabulary, which consists of stop words and common words used in spontaneous lecture speech (in the experimental evaluation described in Section 4 this common vocabulary consisted of 1788 words). In addition for each lecture we then select a topic-specific vocabulary based on a set of initial seed documents, for example lecture-slides, handouts or book chapters. This unsupervised vocabulary selection approach consists of two parts. First the collection of documents related to the lecture at hand and second the ranking and selection of an active recognition vocabulary.

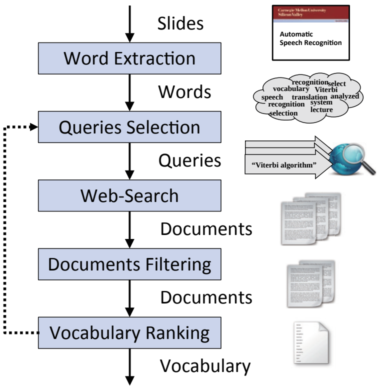
Slides
Word Extraction
Words
Queries Selection
Queries
Web-Search
Documents
Documents Filtering
Documents
Vocabulary Ranking
Vocabulary
Carnegie Mellon University Silicon Valley
Automatic Speech Recognition
recognition select vocabulary Viterbi speech translation analyzed recognition system selection lecture
"Viterbi algorithm"

Figure 3: Document Collection and Vocabulary Ranking Process

### 3.1 Document Collection

Figure 3 illustrates the document collection and vocabulary ranking process. The document collection process begins with a set of lecture slides[1] from which words and key phrases are extracted. Search queries are then generated and a large number of web documents are collected by performing a web-based search (here using the Microsoft Bing search engine). The resulting documents are then filtered. This document collection process is described in detail in the following subsections.

[1] If slides are not available, it should be possible to use a similar seed document which contains textual information on the topic of the lecture

**Word Extraction** The first step in document selection involves extracting text from the lecture slides. Symbols and punctuation are removed and the text is lowercased and split into individual words. The resulting word-list is then verified against an extremely large vocabulary to remove erroneous words that were introduced during the extraction process. In the experimental evaluation described in this paper we used the unigram occurrences from the Google Book Ngrams dataset[2] (Michel et al., 2011) which in total contains 3M word entries.

**Query Selection** Next, search queries are generate from the word sequences extracted from the lecture slides. Single words and phrases of two or three words which do not contain any topic-independent vocabulary are selected as queries.

**Web-Search** Web-search is then performed using this query list. The search is limited to find only results in the source language and for each query, the 50 highest ranked documents were selected. The text from the resulting documents (web page or PDF file) were then extracted using a process similar to that used for the lecture slides.

**Document Filtering** After performing search language identification is performed on the resulting documents to ensure they are in the source language. If the percentage of topic-independent vocabulary $V_{independent}$ in a document is higher than a predefined threshold the document is assumed to be in the source language.

$$\text{Threshold} < \frac{|\{w_i | w_i \in V_{independent}\}|}{|W_d|}, \quad (1)$$

where $w_i \in W_d$ is the word occurrences in document $d$. The $\text{Threshold}$ was selected based on a small set of tuning data and was found to be robust across lectures.

**Vocabulary Ranking** Finally, features (described in section 3.2.1) are calculated for each unique word that occur in the lecture slides or retrieved documents. The resulting vocabulary is then ranked using either the value of a single feature, a linear combination of multiple features, or a gaussian mixture model trained on multiple features.

## 3.2 Vocabulary Selection

### 3.2.1 Features

In the vocabulary selection step features are calculated for each word observed during the retrieval process. Ranking is then performed using either an individual feature or a combination of multiple features.

The definition of the features used in this work follows. In these definitions: $D$ is the set of all documents, $Q$ is the set of all queries, and $W$ is the set of all words. The set which contains all documents which contain the word $w_i$ is $D_{w_i}$ (equation 2). The set which contains all documents which were found by the query $q_k$ is $D_{q_k}$ (equation 3) and the set which contains all queries that found the word $w_i$ is $Q_{w_i}$ (equation 4).

$$D_{w_i} = \{d \in D | w_i \in d\} \quad (2)$$

$$D_{q_k} = \{d \in D | d \in q_k\} \quad (3)$$

$$Q_{w_i} = \{q \in Q | \exists d \in D : w_i \in d \wedge d \in q\} \quad (4)$$

**Document Features** For each document, two similarities metrics between the document and the lecture slides are calculated. These similarities are based on the cosine similarity (equation 5), which calculates the cosine distance between two vectors $\mathbf{a}$ and $\mathbf{b}$ in the following manner:

$$\text{cosine}(\mathbf{a}, \mathbf{b}) = \frac{\sum_{i=1}^{n} \mathbf{a}_i \times \mathbf{b}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{a}_i)^2} \times \sqrt{\sum_{i=1}^{n} (\mathbf{b}_i)^2}} \quad (5)$$

**Cosine Similarity based on Word Frequency** Equation 6 shows the first similarity metric $\text{WFS}(d_k)$ between the slides and the document $d_k$.

$$\text{WFS}(d_k) = \text{cosine}(\mathbf{freq}_{slides}, \mathbf{freq}_{d_k}) \quad (6)$$

where $\mathbf{freq}_{slides}$ is the word frequency vector for the slides and $\mathbf{freq}_{d_k}$ is the word frequency vector for the document $d_k$. The word frequency vector is explained in equation 7.

$$\mathbf{freq}_x = (\text{count}_x(w_1), ..., \text{count}_x(w_n)) \quad (7)$$

where $w_1, ..., w_n$ are the $n$ unique words which occur in the slides, $\text{count}_{slides}(w_i)$ is the number of occurrences of the word $w_i$ in the slides, and $\text{count}_{d_k}(w_i)$ is the number of occurrences of the word $w_i$ in the document $k$.

[2] Available at *http://ngrams.googlelabs.com/datasets*

**Cosine Similarity based on Tf-Idf** The second similarity metric $\mathrm{TIS}(d_k)$ (equation 12) is similar to the first, however instead of the word frequencies, the vectors contain the approximated tf-idf (term frequency × inverse document frequency, equations 8 to 11) of every unique word in the slides. Tf-idf is a common metric used for text retrieval (Salton and Buckley, 1988) and is defined as:

$$\mathrm{tf}(w_i, d_k) = \frac{\mathrm{count}_{d_k}(w_i)}{\sum\limits_{w_j \in d_k} \mathrm{count}_{d_k}(w_j)} \tag{8}$$

$$\mathrm{idf}(w_i) = \log \frac{N}{g(w_i)} \tag{9}$$

where $N$ is the number of volumes in the Google Book Ngrams dataset and $g(w_i)$ is the number of volumes that contain the word $w_i$ in the Google Book Ngrams dataset (Michel et al., 2011).

$$\mathrm{tfidf}(w_i, d_k) = \mathrm{tf}(w_i, d_k) \times \mathrm{idf}(w_i) \tag{10}$$

$$\mathbf{tfidf}_x = (\mathrm{tfidf}(w_1, x), ..., \mathrm{tfidf}(w_n, x)) \tag{11}$$

$$\mathrm{TIS}(d_k) = \mathrm{cosine}(\mathbf{tfidf}_{slides}, \mathbf{tfidf}_{d_k}) \tag{12}$$

**Query Features** Each query $q_k$ has two metrics. The first metric $\mathrm{QWF}(q_k)$ is the average similarity between the slides and each document found by this query based on the word frequency (equation 13). The second metric $\mathrm{QTI}(q_k)$ is the average similarity between the slides and each document found by the query based on tf-idf (equation 14).

$$\mathrm{QWF}(q_k) = \frac{\sum_{d \in q_k} \mathrm{WFS}(d)}{|D_{q_k}|} \tag{13}$$

$$\mathrm{QTI}(q_k) = \frac{\sum_{d \in q_k} \mathrm{TIS}(d)}{|D_{q_k}|} \tag{14}$$

**Word Features** For each word $w_i$, 21 Features ($\mathrm{f}_1(w_i), ..., \mathrm{f}_{21}(w_i)$) are calculated (equations 15 to 24). The majority leverage the document and query features listed above.

1. *DocCount*: Number of documents in which the word occurs.
$$\mathrm{f}_1(w_i) = |D_{w_i}| \tag{15}$$

2. *VocCount*: Number of occurrences in all documents.
$$\mathrm{f}_2(w_i) = \sum_{d \in D} count_d(w_i) \tag{16}$$

3. *tfSum*: Sum of term frequencies:
$$\mathrm{f}_3(w_i) = \sum_{d \in D} \frac{count_d(w)}{\sum\limits_{w_i \in W} count_d(w_i)} \tag{17}$$

4. *tfCosineCount*: Sum of term frequencies weighted by the cosine similarity based on word frequency:
$$\mathrm{f}_4(w_i) = \sum_{d \in D} \mathrm{WFS}(d) \frac{count_d(w)}{\sum\limits_{w_i \in W} count_d(w_i)} \tag{18}$$

5. *tfCosineTfidf*: Sum of term frequencies weighted by the cosine similarity based on tf-idf
$$\mathrm{f}_5(w_i) = \sum_{d \in D} \mathrm{TIS}(d) \frac{count_d(w)}{\sum\limits_{w_i \in W} count_d(w_i)} \tag{19}$$

6. *DocCosineCount*: max, min and average of the document feature WFS of all documents ($D_{w_i}$) in which the word $w_i$ occurs.
$$\mathrm{f}_{6,7,8}(w_i) = \mathrm{WFS}_{\mathrm{max,min,avg}}(D_{w_i}) \tag{20}$$

7. *DocCosineTfidf*: max, min and average of the document feature TIS of all documents ($D_{w_i}$) in which the word $w_i$ occurs.
$$\mathrm{f}_{9,10,11}(w_i) = \mathrm{TIS}_{\mathrm{max,min,avg}}(D_{w_i}) \tag{21}$$

8. *QueryScoreCount*: max, min and average of query feature QWF of all queries ($Q_{w_i}$) that found the word $w_i$.
$$\mathrm{f}_{12,13,14}(w_i) = \mathrm{QWF}_{\mathrm{max,min,avg}}(Q_{w_i}) \tag{22}$$

9. *QueryScoreCount*: max, min and average of query feature QTI of all queries ($Q_{w_i}$) that found the word $w_i$.
$$\mathrm{f}_{15,16,17}(w_i) = \mathrm{QTI}_{\mathrm{max,min,avg}}(Q_{w_i}) \tag{23}$$

10. *GooglebookIDF*: Inverse document frequency based on the Google Book Ngrams dataset (equation 9).
$$\mathrm{f}_{18}(w_i) = \mathrm{idf}(w_i) \tag{24}$$

11. *GoogleBookNgrams*: The word features $\mathrm{f}_{19,20,21}$ are the values match_count, page_count and volume_count from the Google Book Ngrams dataset (Michel et al., 2011).

### 3.2.2 Vocabulary Ranking and Selection

The resulting vocabulary after document collection is too large to be incorporated directly into a speech translation system (in our work we observed vocabularies between 135k and 680k) and thus a smaller vocabulary must be selected. To rank a vocabulary for selection, we compared three approaches: single feature ranking, linear feature combination-based ranking, and ranking using gaussian mixture models. We also compared the relationship of vocabulary size to coverage of the lecture transcripts.

**Single Feature Ranking** One method to select a lecture-specific vocabulary is to sort words by one specific feature (e.g., word occurrence). Based on this ranking words are added to the vocabulary until the desired vocabulary size is reached.

**Linear Feature Combination Ranking** Another approach is to combine two or more features linearly and then sort words based on this multi-feature score.

$$\alpha \times \mathrm{f}_i + (1-\alpha) \times \mathrm{f}_j \qquad (25)$$

**Gaussian Mixture Model Ranking** A third approach is to use Gaussian Mixture Models (GMMs) for vocabulary ranking. In this approach two GMMs were trained, the first on words which occurred in the lecture and the second on words which did not occur in the lecture. For ranking, the difference in the log-likelihood of a word feature vector for each of these GMMs was calculated, as in equation 26, and words were ranked by this value.

$$\log \mathrm{P}_{in}(\mathbf{w}) - \log \mathrm{P}_{out}(\mathbf{w}) \qquad (26)$$

## 4 Experimental evaluation

We evaluated the effectiveness of our proposed unsupervised vocabulary selection method for lecture adaption within our German-English Simultaneous Lecture Translation system (Kolss et al., 2008). The evaluation was performed on 4 lectures (lect1, lect2, lect3, lect4) which were held at Karlsruhe Institute of Technology, Germany in 2009 and 2010. The lectures were on a variety of different topics: Data structures (lect1), machine translation (lect2), mechanics (lect3), and population geography (lect4). Our baseline lecture translation system was trained on lectures on Computer Science, and thus performed especially poorly on lect3 and lect4.

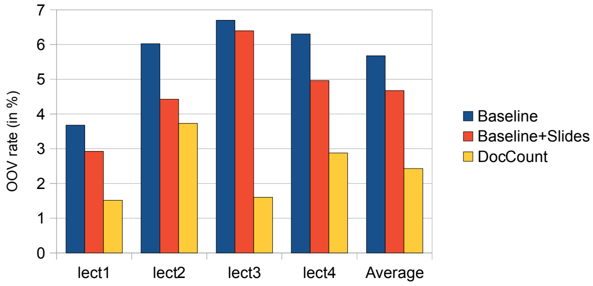

Figure 4: DocCount ranking results for a 40k vocabulary compared with baseline and baseline+slides.

### 4.1 Baseline

Baseline vocabularies with 40k, 90k, and 300k words were selected from a combined corpora of broadcast news, parliamentary debates, printed media, and university web data using the method described in (Stüker et al., 2010). Using these vocabularies, the average OOV rate across the four lectures were: 5.7% (40k), 4.1% (90k), and 3.1% (300k). Adding vocabulary that occurred in the lecture slides ("Baseline+Slides") reduced OOV rate on average by 18.0%, obtaining average OOV rates of 4.7% (40k), 3.3% (90k), and 2.6% (300k). A detailed breakdown per lecture for 40k vocabularies is shown in Figure 4.

### 4.2 Feature-based Vocabulary Selection

First, we selected vocabularies by ranking them by a single feature. The average OOV rate using a 40k vocabularies is shown in figure 5. The lowest OOV rate was obtained using feature 1, DocCount ($\mathrm{f}_1$). The feature 2, VocCount ($\mathrm{f}_2$) obtained a similar OOV rate. Figure 6 shows the OOV rate for lecture 1 using these two features, compared with random vocabulary selection and the baseline 40k, 90k, and 300k vocabularies. The OOV rate for DocCount and VocCount is very similar, and vocabulary selection using these two features is significantly lower than the OOV rate of the three baseline systems. Figure 4 shows the OOV rate of a 40k vocabulary selected using the DocCount feature compared to the Baseline (with and without slides). For all four lectures, the OOV rate is significantly lower than the proposed

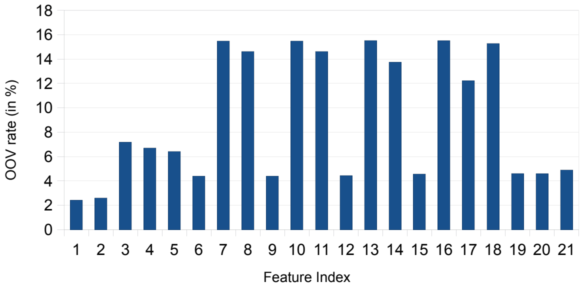

Figure 5: Average OOV rate for all features (40k vocabulary).

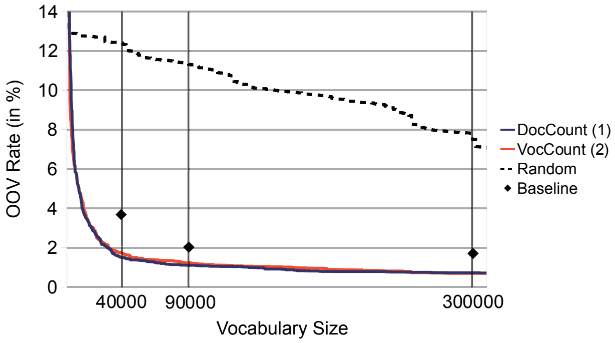

Figure 6: DocCount and VocCount compared to Baseline and Random on lecture 1.

Baseline vocabularies even when slides were added. Using the proposed vocabulary selection with the DocCount feature improved our baseline OOV rate on average by $59.8\%$ while maintaining the same vocabulary size.

## 4.3 Feature Combination

Next, we investigated the effectiveness of combining multiple features for vocabulary ranking. The two approaches we investigated were linear feature combination and the gaussian mixture model method described in section 3.2.2.

### 4.3.1 Linear Combination

We analyzed the effectiveness of linear combining two or more features during vocabulary ranking. We linearly combined pairs of features (equation 25) evaluating across all feature combinations and weights $\alpha \in \{0.1, 0.2, ..., 0.9\}$. We observed that combining DocCount and VocCount with $\alpha = 0.5$ obtained an average reduction of OOV rate of $2.3\%$ compared to using the DocCount feature ($\mathrm{f}_1$) alone.

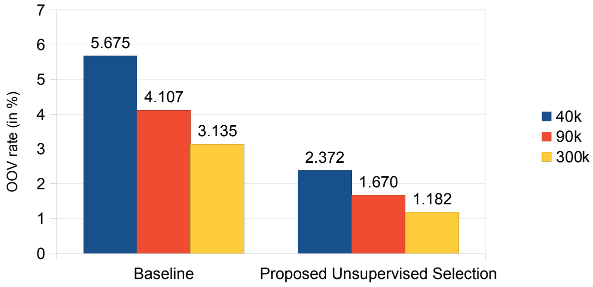

Figure 7: Average OOV rate of baseline compared with linear combination in different vocabulary sizes.

A selection of results for linear combination are shown in Table 1. Figure 7 shows the effectiveness of our proposed linear combination approach using the DocCount feature ($\mathrm{f}_1$) and the VocCount feature ($\mathrm{f}_2$) compared to the baseline over varying vocabulary sizes. On average the OOV rate of our proposed approach with a 40k vocabulary is lower than the OOV rate of the 300k Baseline vocabulary showing the effectiveness of this method.

### 4.3.2 Gaussian Mixture Models

In this approach, we trained gaussian mixture models for all feature pairs using labeled data from one lecture. This models were then used for vocabulary ranking (equation 26). GMMs were trained with two components. Table 1 shows the results for three feature pairs using GMMs with two components. Using the GMM-based combination, the average OOV rate was similar to that obtained using the single feature DocCount ($\mathrm{f}_1$). No feature-pairs consistently improved performance across all lectures.

## 5 Conclusion

Effective adaptation techniques are required to enable lecture transcription and lecture translation systems to perform adequately across diverse lecture topics. Our proposed web-based approach solves one of the key issues in current systems, that of selecting an appropriate topic-specific vocabulary for real-time speech recognition and translation. Using our approach, the OOV rate was reduced by up to $83.8\%$ (on average by $59.9\%$) compared to a baseline vocabulary. We also analyzed two methods to optimize the vocabulary selection using fea-

| Approach | Features | Lecture 1 | Lecture 2 | Lecture 3 | Lecture 4 | Average |
|---|---|---|---|---|---|---|
| baseline | - | 3.680 | 6.021 | 6.697 | 6.303 | 5.675 |
| single | DocCount | 1.514 | 3.730 | 1.598 | 2.876 | 2.429 |
| linear | DocCount (0.5), VocCount (0.5) | 1.496 | 3.663 | 1.446 | 2.885 | 2.372 |
| linear | DocCount (0.8), tfCosineCount (0.2) | 1.470 | 3.763 | 1.370 | 2.901 | 2.376 |
| linear | DocCount (0.6), tfCosineCount (0.4) | 1.514 | 4.240 | 1.294 | 3.008 | 2.514 |
| GMM | DocCount, VocCount | 1.522 | 3.621 | 1.446 | 2.942 | 2.383 |
| GMM | DocCosineCount-Max, -Avg | 1.949 | 3.546 | 1.446 | 3.550 | 2.623 |
| GMM | DocCount, DocCosineTfidf-Avg | 1.618 | 4.039 | 2.283 | 2.901 | 2.710 |

Table 1: Feature Combination - Linear Combination and Gaussian Mixture Models (OOV Rate in %, 40k Vocabularies)

ture combinations. During this tests, we identified a linear combination which leads to a further improvement of 2.3% compared to a single feature. Our results indicate that the quality of the data corpus is more important than the specific selection method, thus, in our future work, we intend to optimize our document retrieval method for vocabulary coverage and incorporate our approach into an online end-to-end lecture translation system.

## Acknowledgments

We would like to thank Sebastian Stueker and Kevin Kilgour for providing the baseline German speech recognition system used in this work as well as their technical expertise throughout the project. This work was supported in part by an interACT student exchange scholarship.

## References

Christian Fügen. 2009. *A System for Simultaneous Translation of Lectures and Speeches*. Phd thesis, University of Karlsruhe.

James R. Glass, Timothy J. Hazen, Lee Hetherington, and Chao Wang. 2004. Analysis and Processing of Lecture Audio Data: Preliminary Investigations. In *Proc. HLT-NAACL*, pages 9–12.

James R. Glass, Timothy J. Hazen, Scott Cyphers, Igor Malioutov, David Huynh, and Regina Barzilay. 2007. Recent Progress in the MIT Spoken Lecture Processing Project. In *Proc. Interspeech*, pages 2553–2556.

Tatsuya Kawahara, Yusuke Nemoto, and Yuya Akita. 2008. Automatic Lecture Transcription by Exploiting Presentation Slide Information for Language Model Adaptation. In *Proc. ICASSP*, pages 4929–4932.

Tatsuya Kawahara. 2010. Automatic Transcription of Parliamentary Meetings and Classroom Lectures. In *Proc. ISCSLP*, pages 1–6.

Muntsin Kolss, Matthias Wölfel, Florian Kraft, Jan Niehues, Matthias Paulik, and Alex Waibel. 2008. Simultaneous German-English Lecture Translation. In *Proc. IWSLT*, pages 174–181.

Jean-Baptiste Michel, Yuan Kui Shen, and Aviva Presser Aiden. 2011. Quantitative Analysis of Culture Using Millions of Digitized Books. *Science*, 331:176–182, December.

Cosmin Munteanu, Gerald Penn, and Ron Baecker. 2007. Web-based Language Modelling for Automatic Lecture Transcription. In *Proc. Interspeech*, number August, pages 2353–2356.

Alex Park, Timothy J. Hazen, and James R. Glass. 2005. Automatic Processing of Audio Lectures for Information Retrieval: Vocabulary Selection and Language Modeling. In *Proc. ICASSP*, volume 1, pages 497–500.

Gerard Salton and Christopher Buckley. 1988. Term-Weighting Approaches in Automatic Text Retrieval. *Information Processing & Management*, 24(5):513–523.

Hagen Soltau, Florian Metze, Christian Fügen, and Alex Waibel. 2001. A one-pass decoder based on polymorphic linguistic context assignment. In *Proc. ASRU*, pages 214–217.

Sebastian Stüker, Kevin Kilgour, and Jan Niehues. 2010. Quaero Speech-to-Text and Text Translation Evaluation Systems. In *Proc. HLRS*, pages 529–542. Springer.

Stephan Vogel, Ying Zhang, Fei Huang, Alicia Tribble, Ashish Venugopal, Bing Zhao, and Alex Waibel. 2003. The CMU statistical machine translation system. In *Proc. MT Summit IX*, volume 9, page 54.

Stephan Vogel. 2003. Using Noisy Bilingual Data for Statistical Machine Translation. In *Proc. EACL*, pages 175–178.

Hiroki Yamazaki, Koji Iwano, Koichi Shinoda, Sadaoki Furui, and Haruo Yokota. 2007. Dynamic Language Model Adaptation Using Presentation Slides for Lecture Speech Recognition. In *Proc. Interspeech*, pages 2349–2352.