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 students 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 lan-
guage translation system and the topic of conversa-
tion is severe, vocabulary coverage is poor lead-
ing to a high number of out-of-vocabulary (OOV)
words, poor translation quality and low intelligibil-
ity. For effective adaptation vocabulary coverage is
a key component that prior works have often over-
looked.

In (Kolss et al., 2008), a system for translating
German lectures into English was introduced. They
selected the system vocabulary based on word oc-
currence counts in both in-domain (lecture transcrip-
tions, presentation slides and web data) and out-of-
domain corpora and built lecture-independent mod-
els for speech recognition and machine translation
using these corpora. (Munteanu et al., 2007) in-
troduced an approach for language model adapta-
tion 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 im-
proved 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 (Ya-
mazaki et al., 2007) in which words from the lecture
slides were first added to the active system vocab-
ulary and then language model adaptation was per-
formed 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; Kawa-
hara, 2010) with an additional step in which lan-
guage model adaptation was performed independ-
ently for each slide, resulting in an adaptive lan-
guage model which followed the course of the on-
gothing lecture. Within the MIT Spoken Lecture Pro-
cessing Project (Glass et al., 2007) a lecture-specific
vocabulary was extracted from manually provided
supplemental text provided by the lecturer, includ-
ing lecture slides, journal articles, and book chap-
ters, 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 cov-
ention. 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 com-
pared to using topic specific vocabularies. In this
work, we propose a novel approach to improve vo-
cabulary coverage based on a feature-based vocabu-
lary ranking scheme and documents collected from
the WWW. Our proposed approach significantly im-
proves vocabulary coverage compared to a lecture-
independent system and further improves the effec-
tiveness of other adaptation approaches including
both language model adaptation for speech recogni-
tion (Munteanu et al., 2007) and adaptation of ma-
cine translation models based on comparable cor-
pora (Vogel, 2003).

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 Uni-
versity (USA). This system, illustrated in Figure 1,
simultaneously translates lectures in real-time from
the speaker’s language into multiple languages re-
quired 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 speak-
ers language and want to have supplemental lan-
guage assistance. Spoken translation output can be
listened to either via headphones or targeted audio
speakers, which make it possible to send the trans-
Figure 2: Components of the Lecture Translation System

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). Input speech from the lecturer is recognized by the ASR component (Soltan 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 (SMT) 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. For each lecture the speech recognition dictionary, translation model, and both source and target language models should be adapted to the specific lecture topic. The active vocabulary in the source language is critical because this defines the vocabulary used in the end-to-end system.

3 Web-Search based Vocabulary Adaptation

The vocabulary used in 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.

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 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.

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 dictionary 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 (Michel et al., 2011) which contains a total of 3M word entries.

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

2Available at http://ngrams.googlelabs.com/datasets
Query Selection  Next, search queries are generated 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 in a document is higher than a predefined threshold the document is assumed to be in the source language.

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

where \( w_i \in W_d \) is the word occurrences in document \( d \). The Threshold was selected based on a small set of tuning data and was found to be robust across languages.

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 that found the word \( w_i \) 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).

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

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

\[
Q_{w_i} = \{q \in Q | \exists d \in D : w_i \in d \land d \in q\} \quad (4)
\]

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 similarity \( \text{cosine}(a, b) \) between two vectors \( a \) and \( b \) in the following manner:

\[
\text{cosine}(a, b) = \frac{\sum_{i=1}^{n} a_i \times b_i}{\sqrt{\sum_{i=1}^{n} (a_i)^2} \times \sqrt{\sum_{i=1}^{n} (b_i)^2}} \quad (5)
\]

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

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

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

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

where \( w_1, ..., w_n \) are the \( n \) unique words which occur in the slides, \( \text{count}_{\text{slides}}(w_i) \) 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 \( k \).

Cosine Similarity based on Tf-Idf  The second similarity metric \( TIS(d_k) \) (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:

\[
tf(w_i, d_k) = \frac{\text{count}_{d_k}(w_i)}{\sum_{w_j \in d_k} \text{count}_{d_k}(w_j)} \quad (8)
\]
\[
\text{idf}(w_i) = \log \frac{N}{g(w_i)} \tag{9}
\]
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).

\[
tfidf(w_i, d_k) = \text{tf}(w_i, d_k) \times \text{idf}(w_i) \tag{10}
\]
\[
tfidf_x = (\text{tfidf}(w_1, x), \ldots, \text{tfidf}(w_n, x)) \tag{11}
\]
\[
\text{TIS}(d_k) = \text{cosine}(\text{tfidf}_{\text{slides}}, \text{tfidf}_{d_k}) \tag{12}
\]

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

\[
\text{QWF}(q_k) = \frac{\sum_{d \in q_k} \text{WFS}(d)}{|D_{q_k}|} \tag{13}
\]
\[
\text{QTI}(q_k) = \frac{\sum_{d \in q_k} \text{TIS}(d)}{|D_{q_k}|} \tag{14}
\]

**Word Features** For each word \(w_i\), 21 Features \((f_1(w_i), \ldots, f_{21}(w_i))\) 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.
   \[
f_1(w_i) = |D_{w_i}| \tag{15}
\]
2. **VocCount**: Number of occurrences in all documents.
   \[
f_2(w_i) = \sum_{d \in D} \text{count}_{d}(w_i) \tag{16}
\]
3. **tfSum**: Sum of term frequencies:
   \[
f_3(w_i) = \sum_{d \in D} \frac{\text{count}_{d}(w)}{\sum_{w_i \in W} \text{count}_{d}(w_i)} \tag{17}
\]
4. **tfCosineCount**: Sum of term frequencies weighted by the cosine similarity based on word frequency:
   \[
f_4(w_i) = \sum_{d \in D} \text{WFS}(d) \frac{\text{count}_{d}(w)}{\sum_{w_i \in W} \text{count}_{d}(w_i)} \tag{18}
\]
5. **tfCosineTfidf**: Sum of term frequencies weighted by the cosine similarity based on tf-idf
   \[
f_5(w_i) = \sum_{d \in D} \frac{\text{TIS}(d)}{\sum_{w_i \in W} \text{count}_{d}(w_i)} \tag{19}
\]
6. **DocCosineCount**: \(\max, \min, \text{and average}\) of the document feature \(\text{WFS}(d)\) of all documents \((D_{w_i})\) in which the word \(w_i\) occurs.
   \[
f_{6,7,8}(w_i) = \text{WFS}_{\max,\min,\text{avg}}(D_{w_i}) \tag{20}
\]
7. **DocCosineTfidf**: \(\max, \min, \text{and average}\) of the document feature \(\text{TIS}(d)\) of all documents \((D_{w_i})\) in which the word \(w_i\) occurs.
   \[
f_{9,10,11}(w_i) = \text{TIS}_{\max,\min,\text{avg}}(D_{w_i}) \tag{21}
\]
8. **QueryScoreCount**: \(\max, \min, \text{and average}\) of query feature \(\text{QWF}(q_k)\) of all queries \((Q_{w_i})\) that found the word \(w_i\).
   \[
f_{12,13,14}(w_i) = \text{QWF}_{\max,\min,\text{avg}}(Q_{w_i}) \tag{22}
\]
9. **QueryScoreCount**: \(\max, \min, \text{and average}\) of query feature \(\text{QTI}(q_k)\) of all queries \((Q_{w_i})\) that found the word \(w_i\).
   \[
f_{15,16,17}(w_i) = \text{QTI}_{\max,\min,\text{avg}}(Q_{w_i}) \tag{23}
\]
10. **GooglebookIDF**: Inverse document frequency based on the Google Book Ngrams dataset (equation 9).
    \[
f_{18}(w_i) = \text{idf}(w_i) \tag{24}
\]
11. **Google Book Ngrams**: The word features \(f_{19,20,21}\) 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 vocabulary size after document collection is usually too large to be incorporated directly into a speech translation system (in our work we observed vocabularies between 135k and 680k). The resulting vocabulary thus must be ranked based on relevance to the lecture topic at hand and a final vocabulary...
of a much smaller size must be selected for the end-to-end speech translation system. For ranking, we compared three different 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 over the lecture transcripts.

**Single Feature Ranking**  One way to select the lecture-specific vocabulary is to sort the 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**  An additional approach is to combine two or more features linearly and then sort words based on this multi-feature combination.

\[ \alpha \times f_i + (1 - \alpha) \times f_j \]  

(25)

**Gaussian Mixture Model Ranking**  The third approach we investigated used Gaussian Mixture Models (GMMs) for vocabulary ranking. 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, equation 26, and words were ranked by this value.

\[ \log P_{in}(w) - \log P_{out}(w) \]  

(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.

#### 4.1 Baseline

Before the evaluation of our system, we determined the baseline out-of-vocabulary (OOV) rate with three vocabularies with 40k, 90k, and 300k words. These vocabularies 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). As a simple adaptation step, we added the words from the slides to the vocabulary. The results are shown in the Figures 5, 6, and 7 compared with vocabularies described in the next sections (section 4.2 and 4.3). The average OOV rate on the four lectures for the three baseline vocabularies (40k, 90k, and 300k) are: 5.675% (40k), 4.107% (90k), and 3.135% (300k). As expected, the OOV rate is lower with a larger vocabulary. Adding the slides to the vocabulary improves the OOV rate on average by 18.0%.

#### 4.2 Single Feature Comparison

The first method of selecting the vocabulary is for each of the 21 features to sort the words descending by the feature and then use the first words for the vocabulary. The result of our experiments were very different depending on the feature we used for the vocabulary selection. The average OOV rate for 40k vocabularies built using the 21 features is shown in figure 4. The best results were obtained using the feature 1, DocCount \( (f_1) \). The second best feature was 2, VocCount \( (f_2) \). Figure 5 shows the result for the features DocCount and VocCount compared with random and the baseline on lecture 1. The graph shows that the OOV rate for DocCount and VocCount is very similar. The OOV rate for both features is lower than the OOV rate of the three baseline systems. Figure 6 shows the OOV rate results of 40k DocCount vocabulary compared to the 40k Baseline vocabulary (with and without slides). For all four lectures, the OOV rate is lower than the 40k Baseline vocabulary with added slides. Using the web-search based vocabulary adaptation with DocCount ranking improves 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

Next, we analyzed if we can improve the OOV rate by linear combination of two or more features. We linearly combined two features (equation 25) and tested this ranking approach for all feature combinations and all \( \alpha \in \{0.1, 0.2, ..., 0.9\} \). For some linear combinations the resulting vocabulary had a better OOV rate than the results from using a single feature. But the same combination which has a better result in one lecture often led to a worse result in another lecture. The combination of DocCount and VocCount with the weights 0.5 receives a better result for three of the four lectures, with a minimal increase in OOV for lecture 4 (an increased of 0.009 percent points). Using this feature combination strategy the average OOV rate improved by 2.3% compared to the case when the DocCount feature was used alone. A selection of results are shown in Table 1 and Figure 7 shows the effectiveness of our proposed linear combination approach (DocCount and VocCount) compared to the baseline over different 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 strength of this method.

![Figure 4: Average OOV rate for all features (40k vocabulary).](image)

![Figure 5: DocCount and VocCount compared to Baseline and Random on lecture 1.](image)

![Figure 6: DocCount ranking results for a 40k vocabulary compared with baseline and baseline+slides.](image)

![Figure 7: Average OOV rate of baseline compared with linear combination in different vocabulary sizes.](image)

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

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

We trained gaussian mixture models for all feature pairs on labeled data of one lecture and used this GMMs to select the vocabulary for the same lecture using equation 26. We trained GMMs with one, two, three, and four components. Similar to the results from the linear combination, the GMM approach gives better results on one lecture for some feature pairs but the same feature pair has a worse result on another lecture. None of our tested feature pairs has a better result for all lectures. The results of the GMMs with a different number of components on the same feature pair are very similar. Table 1 shows the results for three feature pairs using GMMs with two components. The average OOV rate is around the OOV rate of the best single feature approach. This indicates that the used word features are not sufficient to effectively separate the words which are used in the lecture from the words which are not used in the lecture using GMMs.

5 Conclusion

Current lecture transcription and lecture translation systems need a good and cheap adaptation method to be usable for diverse lecture topics. Our web-search based vocabulary adaptation approach solves one of the key issues in current systems. Using our approach, the OOV rate improved significantly by up to 83.8% (on average by 59.9%) compared to our baseline vocabularies. We also analyzed two methods to optimize the vocabulary selection using feature combinations. During this tests, we identified a linear combination which leads to a further improvement of 2.3% compared to a single feature. Although, there is a small improvement simply counting in how many documents a word occurs or how often it occurs in the corpus is still a very good approach. Our results indicate that the quality of the data corpus is more important than the specific selection method. In our future work, we intend to optimize our document retrieval method for vocabulary coverage and incorporate our approach into an n-to-n system.

References