Unsupervised Vocabulary Selection for Simultaneous Lecture Translation

Paul Maergner[1,2]*, Kevin Kilgour*[2]*, Ian Lane*[1]*, Alex Waibel*[1,2]

[1] Carnegie Mellon University, USA
[2] Karlsruhe Institute of Technology, Germany

paul.maergner@sv.cmu.edu, kevin.kilgour@kit.edu, lane@cs.cmu.edu, waibel@cs.cmu.edu

Abstract

In this work, we propose a novel method for vocabulary selection which enables simultaneous speech recognition systems for lectures to automatically adapt to the diverse topics that occur in educational and scientific 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 a lecture-specific vocabulary and language model based on the resulting documents. In this paper, we introduce a novel method for vocabulary selection where we rank vocabulary that occurs in the collected documents based on a relevance score which is calculated using a combination of word features. Vocabulary selection is a critical component for topic adaptation that has typically been overlooked in prior works. On the interACT German-English simultaneous lecture translation system our proposed approach significantly improved vocabulary coverage, reducing the out-of-vocabulary rate on average by 57.0% and up to 84.9%, compared to a lecture-independent baseline. Furthermore, our approach reduced the word error rate by up to 25.3% (on average 13.2% across all lectures), compared to a lecture-independent baseline.

1. Introduction

Education is increasingly becoming a 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 have thus investigated the use of speech-translation technologies to translate lectures in real-time [1]. The biggest downfall of these systems however is portability. Current systems 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.

In modern simultaneous speech translation systems such as the interACT simultaneous lecture translation system described in [1], speech recognition is performed by applying search across three models, an acoustic model, which models the phonetic units in the input language, a language model (LM), which models the likelihood of word sequences, and a recognition vocabulary, which models the pronunciation of individual words. To allow real-time processing, the size of the recognition vocabulary must be limited, typically in the range of 30k-60k words. Words not present in active system vocabulary will not be be recognized correctly and will often lead to additional errors to the surrounding content. When the mismatch between the training data used to build the ASR system and the topic of the lecture is severe, vocabulary coverage is poor leading to a high number of out-of-vocabulary (OOV) words, low recognition accuracy and low intelligibility in the resulting transcript. To ensure high intelligibility and faster than real-time processing, the selection of an appropriate vocabulary for simultaneous speech translation is critical.

To determine the importance of vocabulary selection for this task, we performed a set of exploratory experiments in which we limited the recognition vocabulary applied during speech recognition to only those words uttered during a specific lecture. These ”oracle” vocabularies significantly improved both speech recognition accuracy and processing speed. On average word error rate (WER) was reduced by 9.7 points (30.2% relative reduction) compared to an unadapted system and decoding time was more than halved. Interestingly, the difference in vocabulary between individual lectures was large. Only 5% of vocabulary (on average 47% of the spoken words per lecture) was common across all six lectures we evaluated on. Although vocabulary selection is a key component for effective adaptation, it has often been overlooked by prior works.

There have been a number of recent works that have proposed methods to deal with the diversity of topics encountered in lecture speech. In [1], 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. The development of lecture-independent models was the goal of this work and no lecture-specific adaptation was performed. Munteanu et al. [2] introduced an approach for language model adaptation which leveraged 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 [3] 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 [2]. A similar approach was applied for automatic subtitling of lectures for the hearing impaired in [4] 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 [5] a lecture-specific vocabulary was extracted from supplemental text provided by the lecturer, including lecture slides, journal articles, and book chapters, which were made available prior to the lecture.

Although the adaptation approaches described above were effective for language model adaptation they did not significantly improve vocabulary coverage. Even when all words that occurred in the lecture slides were added to the active vocabulary, the out-of-vocabulary rate remained 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 applied on documents automatically collected from the WWW. Our proposed approach improves vocabulary coverage, LM perplexity, and speech recognition accuracy compared to a lecture-independent system and further improves the effectiveness of other adaptation approaches including both language model adaptation for speech recognition [2] and possibly the adaptation of machine translation using comparable corpora [6].

2. The interACT Simultaneous Lecture Translation System

The interACT Simultaneous Lecture Translation System [1] 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 the speaker's language and want to have supplemental language assistance. Spoken translation output can be listened to either via headphones or targeted audio speakers, which make it possible to send the translated audio only to a small group of people while the other listeners are not disturbed.

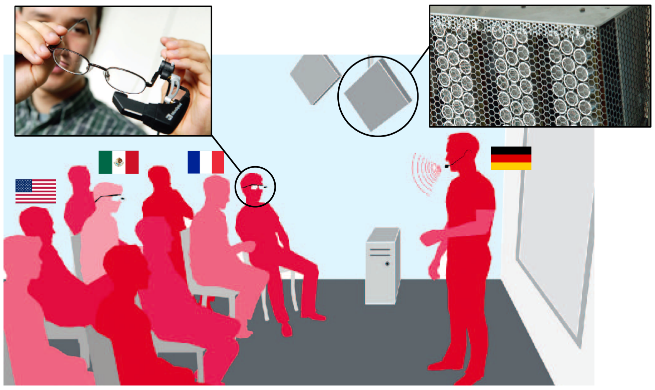

Figure 1: The interACT Lecture Translation System.

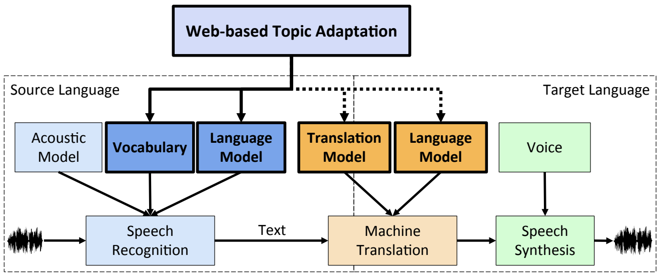

Figure 2: Components of the Lecture Translation System

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 [7] and the resulting output is segmented into sentence-like units which are then passed to MT. The resulting segments are then translated into one or more target languages via our statistical machine translation (SMT) engine STTK [8]. The translated text is either directly displayed to attendees or optionally converted

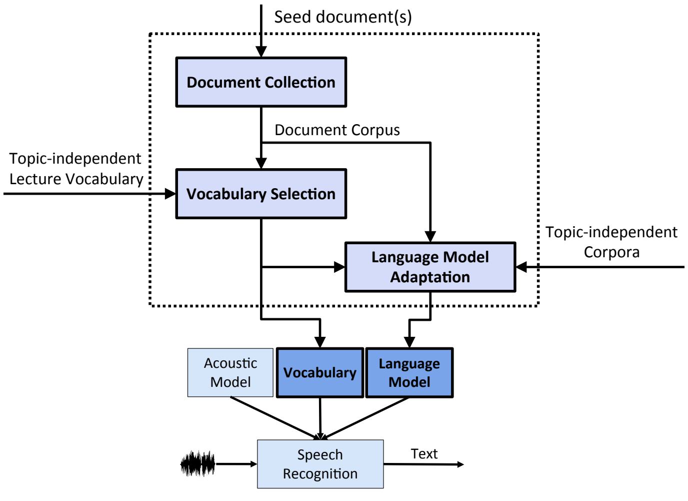

Figure 3: Unsupervised Vocabulary Selection and Language Model Adaptation

into speech output using a TTS engine. For each lecture the speech recognition vocabulary, 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. Unsupervised Vocabulary Selection and Language Model Adaptation

The vocabulary used by a presenter during a lecture can be seen as a combination of two vocabularies as described in [5]: 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 our common vocabulary consisted of 1788 words). In addition to this vocabulary, we then select a topic-specific vocabulary for each lecture based on a set of initial seed documents, for example lecture-slides, handouts, or book chapters. Using these seed documents, our proposed system automatically collects a large corpus of related documents from the World Wide Web and then selects an active recognition vocabulary using a feature-based word ranking computed using this corpus. Additionally, the document corpus is used to adapt the language model to the topic of the lecture. The whole adaptation process consists of three steps, document collection, vocabulary selection, and language model adaptation. Figure 3 illustrates the three steps of the adaptation process, which are described in detail in the following subsections.

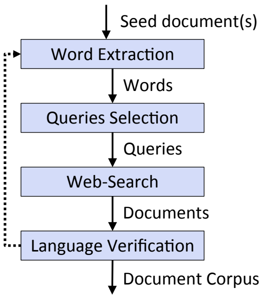

Figure 4: Document Collection

3.1. Document Collection

Figure 4 illustrates the document collection process. The document collection process begins with one or more seed documents, such as the slides of the lecture, 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-search. Then, language verification is performed on the resulting documents. The document collection process is described in detail in the following.

1. **Word Extraction:** The first step in document selection involves extracting text from the seed documents. 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 are introduced during the extraction process. In the experimental evaluation described in this paper, we used the unigram occurrences from the Google Book Ngrams dataset[1] [9], which in total contains 3M word entries.

2. **Query Selection:** Next, search queries are generated from the lecture slides. Here, short phrases of up to three words which do not contain any topic-independent vocabulary are selected as search queries.

3. **Web-Search:** Web-search[2] 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. Then, the text from the resulting documents (web page or PDF file) is extracted.

4. **Language Verification:** For each document, language verification is performed to ensure that it is actually in the required language. When the percentage of topic-independent vocabulary in the document is below 30%, the document is removed from further processing.

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

[2] Search is performed using the Microsoft Bing search engine.

3.2. Vocabulary Selection using Feature-based Ranking

After document collection, the resulting vocabulary is too large to be incorporated directly into an ASR system (in our work we observed vocabularies between 135k and 850k) and thus a smaller active recognition vocabulary must be selected. To select words for this smaller vocabulary, a ranking score for each word is computed. Words with the highest score are added to the vocabulary until the desired vocabulary size is reached. The *ranking score* is based on the different word features described in section 3.3. We compared different scoring functions $s(w)$ to compute the ranking of each word w based on its specific word features $f_i(w)$:

1. **Single Feature Score:** The score $s_{single,i}(w)$ is based on one single feature $f_i(w)$ (e.g. DocCount):

$$s_{single,i}(w) = f_i(w) \tag{1}$$

2. **Linear Feature Combination Score:** The score $s_{linear}(w)$ is defined as a linear weighting of two or more features. For example:

$$s_{linear,i,j}(w) = \alpha \cdot f_i(w) + (1-\alpha) \cdot f_j(w) \tag{2}$$

3. **Gaussian Mixture Model Score:** The score $s_{gmm}(w)$ is based on the likelihood ratio of two Gaussian Mixture Models (GMMs). Two GMMs are trained, one on words which occur in a specific lecture and one on words which do not occur. The score $s_{gmm}(w)$ is the difference in the log-likelihood of a word feature vector for each of these GMMs. For example with the word feature vector $\mathbf{f}_{i,j}(w) = \begin{pmatrix} f_i(w) & f_j(w) \end{pmatrix}^T$:

$$s_{gmm,i,j}(w) = \log \mathrm{P}_{in}(\mathbf{f}_{i,j}(w)) - \log \mathrm{P}_{out}(\mathbf{f}_{i,j}(w)) \tag{3}$$

3.3. Features for Vocabulary Selection

The vocabulary ranking scores, described in section 3.2, rely on the features defined in this section. 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 4). The set which contains all documents which were found by the query q_k is D_{q_k} (equation 5) and the set which contains all queries that found the word w_i is Q_{w_i} (equation 6).

$$D_{w_i} = \{d \in D | w_i \in d\} \tag{4}$$

$$D_{q_k} = \{d \in D | d \in q_k\} \tag{5}$$

$$Q_{w_i} = \{q \in Q | \exists d \in D : w_i \in d \land d \in D_q\} \tag{6}$$

3.3.1. 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 7), which has been found to be effective in information retrieval. We are using a simplified version of the cosine similarity which only compares the words which occur in the slides. This modification speeds up the calculation and we believe it has little effect on the final result. The cosine similarity calculates the cosine distance between two vectors $\mathbf{a} = \begin{pmatrix} a_1 & a_2 & \dots & a_n \end{pmatrix}^T$ and $\mathbf{b} = \begin{pmatrix} b_1 & b_2 & \dots & b_n \end{pmatrix}^T$ in the following manner:

$$\mathrm{cosine}(\mathbf{a}, \mathbf{b}) = \frac{\sum_{i=1}^{n} a_i \cdot b_i}{\sqrt{\sum_{i=1}^{n} (a_i)^2} \cdot \sqrt{\sum_{i=1}^{n} (b_i)^2}} \tag{7}$$

We are using a simplified version of the cosine similarity

1. **Cosine Similarity based on Word Frequency:** Equation 8 shows the first similarity metric $\mathrm{WFS}(d_k)$ between the slides s and the document d_k.

$$\mathrm{WFS}(d_k) = \mathrm{cosine}(\mathbf{freq}_s, \mathbf{freq}_{d_k}) \tag{8}$$

where $\mathbf{freq}_s$ is the word frequency vector for the slides s and $\mathbf{freq}_{d_k}$ is the word frequency vector for the document d_k, both for all the words in the slides. The word frequency vector for any document x is explained in detail in equation 9.

$$\mathbf{freq}_x = \begin{pmatrix} \mathrm{count}_x(w_1) & \dots & \mathrm{count}_x(w_n) \end{pmatrix}^T \tag{9}$$

where $w_1, ..., w_n$ are all unique words which occur in the slides, $\mathrm{count}_x(w_i)$ is the number of occurrences of the word w_i in document x.

2. **Cosine Similarity based on Tf-Idf:** The second similarity metric $\mathrm{TIS}(d_k)$ (equation 14) is similar to the first, however instead of the word frequencies, the vectors contain the approximated tf-idf (term frequency · inverse document frequency, equations 10 to 13) of every unique word in the slides. Tf-idf is a common metric used for information retrieval [10] and we are using the following definition:

$$\mathrm{tf}(w_i, d_k) = \frac{\mathrm{count}_{d_k}(w_i)}{\sum_{w_j \in d_k} \mathrm{count}_{d_k}(w_j)} \tag{10}$$

$$\mathrm{idf}(w_i) = \log \frac{N}{g(w_i)} \tag{11}$$

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 [9].

$$\mathrm{tfidf}(w_i, d_k) = \mathrm{tf}(w_i, d_k) \cdot \mathrm{idf}(w_i) \tag{12}$$

$$\mathbf{tfidf}_x = \begin{pmatrix} \mathrm{tfidf}(w_1, x) & \dots & \mathrm{tfidf}(w_n, x) \end{pmatrix}^T \tag{13}$$

$$\mathrm{TIS}(d_k) = \mathrm{cosine}(\mathbf{tfidf}_s, \mathbf{tfidf}_{d_k}) \tag{14}$$

3.3.2. *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 15). 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 16).

$$\mathrm{QWF}(q_k) = \frac{\sum_{d \in q_k} \mathrm{WFS}(d)}{|D_{q_k}|} \tag{15}$$

$$\mathrm{QTI}(q_k) = \frac{\sum_{d \in q_k} \mathrm{TIS}(d)}{|D_{q_k}|} \tag{16}$$

3.3.3. *Word Features*

For each word w_i, 21 Features $(\mathrm{f}_1(w_i), ..., \mathrm{f}_{21}(w_i))$ are calculated (equations 17 to 26). The majority leverage the document and query features listed above.

1. **DocCount**: Number of documents in which the word occurs.
$$\mathrm{f}_1(w_i) = |D_{w_i}| \tag{17}$$
2. **VocCount**: Number of occurrences in all documents.
$$\mathrm{f}_2(w_i) = \sum_{d \in D} count_d(w_i) \tag{18}$$
3. **tfSum**: Sum of term frequencies:
$$\mathrm{f}_3(w_i) = \sum_{d \in D} \frac{count_d(w)}{\sum\limits_{w_i \in W} count_d(w_i)} \tag{19}$$
4. **tfCosineCount**: Sum of term frequencies weighted by the cosine similarity based on word frequency:
$$\mathrm{f}_4(w_i) = \sum_{d \in D} \mathrm{WFS}(d) \frac{count_d(w)}{\sum\limits_{w_i \in W} count_d(w_i)} \tag{20}$$
5. **tfCosineTfidf**: Sum of term frequencies weighted by the cosine similarity based on tf-idf
$$\mathrm{f}_5(w_i) = \sum_{d \in D} \mathrm{TIS}(d) \frac{count_d(w)}{\sum\limits_{w_i \in W} count_d(w_i)} \tag{21}$$
6. **DocCosineCount**: max, min and average of the document feature WFS of all documents (D_{w_i}) in which the word w_i occurs.
$$\mathrm{f}_{6,7,8}(w_i) = \mathrm{WFS}_{\mathrm{max,min,avg}}(D_{w_i}) \tag{22}$$
7. **DocCosineTfidf**: max, min and average of the document feature TIS of all documents (D_{w_i}) in which the word w_i occurs.
$$\mathrm{f}_{9,10,11}(w_i) = \mathrm{TIS}_{\mathrm{max,min,avg}}(D_{w_i}) \tag{23}$$
8. **QueryScoreCount**: max, min and average of query feature QWF of all queries (Q_{w_i}) that found the word w_i.
$$\mathrm{f}_{12,13,14}(w_i) = \mathrm{QWF}_{\mathrm{max,min,avg}}(Q_{w_i}) \tag{24}$$
9. **QueryScoreTfidf**: max, min and average of query feature QTI of all queries (Q_{w_i}) that found the word w_i.
$$\mathrm{f}_{15,16,17}(w_i) = \mathrm{QTI}_{\mathrm{max,min,avg}}(Q_{w_i}) \tag{25}$$
10. **GoogleBookIDF**: Inverse document frequency based on the Google Book Ngrams dataset (equation 11).
$$\mathrm{f}_{18}(w_i) = \mathrm{idf}(w_i) \tag{26}$$
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 [9].

3.4. Language Model Adaptation

Once an active vocabulary has been selected, we adapt the language model (LM) to be applied during recognition using an approach similar to [2]. First, we train a lecture-independent LM using large lecture-independent corpora. Then, for each lecture we train a separate LM using the lecture slides and the resulting web documents found with our document collection approach (section 3.1). A lecture-specific LM is subsequently generated by interpolating these two LMs using the SRILM [11] toolkit. We used a fixed interpolation weights of 0.5 in our experimental evaluation. Kneser-Ney smoothing [12] was applied.

4. Experimental Evaluation

We evaluated the effectiveness of the proposed method on the German speech recognition component in our German-English Simultaneous Lecture Translation system [1]. The evaluation was performed on six lectures held at Karlsruhe Institute of Technology, in 2009 and 2010. The lectures consisted of a variety of topics: Data structures (Lect1), machine translation (Lect2), mechanics (Lect3), population geography (Lect4), computer architecture (Lect5), and copyright law (Lect6).

4.1. Vocabulary Selection

First, we evaluated our proposed vocabulary selection approach in terms of the reduction in out-of-vocabulary (OOV) rate it could provide. Evaluation was performed using only Lectures 1-4, as transcripts of Lectures 5 and 6 were not available when this evaluation took place.

4.1.1. *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

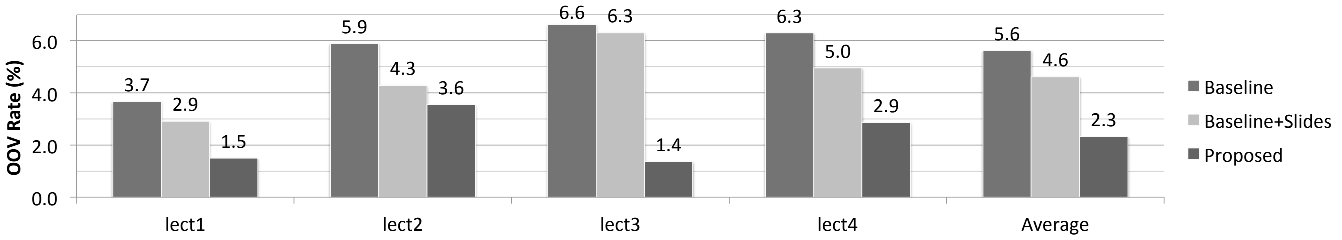

Figure 5: Proposed linear ranking results for a 40k vocabulary compared with baseline and baseline+slides.

using the method described in [13]. Using these vocabularies, the average OOV rate across the four lectures were: 5.6% (40k), 4.0% (90k), and 3.0% (300k). Adding vocabulary that occurred in the lecture slides (”Baseline+Slides”) reduced OOV rate on average by 18.2%, obtaining average OOV rates of 4.6% (40k), 3.2% (90k), and 2.5% (300k). A detailed breakdown per lecture for 40k vocabularies is shown in Figure 5.

4.1.2. Feature-based Vocabulary Selection

First, we selected vocabularies by ranking them by a single feature. The average OOV rate of 40k vocabularies selected using the single feature scores of all 21 features is shown in figure 6. The lowest OOV rate using single feature ranking score was obtained using feature 1, DocCount (f_1), delivering average OOV rates of 2.4% (40k), 1.6% (90k), and 1.1% (300k). The feature 2, VocCount (f_2), obtained similar OOV rates, on average 2.6% (40k), 1.7% (90k), and 1.1% (300k). Vocabulary selection using either of these two features leads to a significantly lower OOV rate than the OOV rate of the three baseline systems. Figure 5 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 Baseline vocabularies even when slides were added. Using the proposed vocabulary selection with the DocCount feature improved our baseline OOV rate on average by 56.8% while maintaining the same vocabulary size.

Next, we investigated the effectiveness of combining multiple features for vocabulary ranking. We linearly combined pairs of features using the linear feature score (section 3.2, eq. 2) evaluating across all feature combinations. We observed that combining DocCount and VocCount with $\alpha = 0.5$ (”Doc+VocCount”) obtained an small average reduction of OOV rate of 1% compared to using the DocCount feature alone, obtaining average OOV rates of 2.3% (40k), 1.6% (90k), and 1.1% (300k). The largest relative reduction in OOV rate was 84.9% which was obtained on lecture 3 for a 300k vocabulary, reducing the OOV rate from 5.0% (Baseline) to 0.8% (Doc+VocCount). GMM-based word ranking (section 3.2, eq. 3) did not reduce the OOV rate compared to the linear case. We evaluated all feature-pairs and although slight improvements were gained for specific lectures no feature-pairs consistently improved performance across all lectures. Fig. 7 shows the effectiveness of our proposed linear score Doc+VocCount compared to the baseline over varying vocabulary sizes. The proposed approach reduced the OOV rate by 58.2%, 55.1%, and 57.7% for the 40k, 90k, and 300k systems. More significantly the 40k vocabulary selected with the proposed approach obtained a lower OOV rate of the 300k Baseline system, showing the effectiveness of this approach.

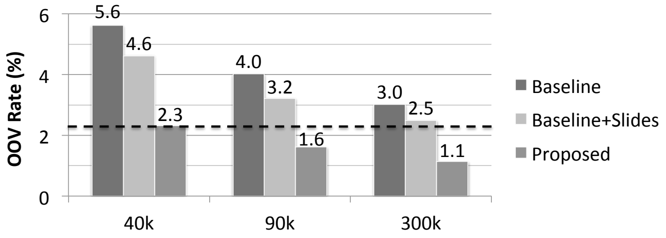

Figure 7: Average OOV rate of baseline compared with linear combination in different vocabulary sizes.

4.2. Lecture-dependent Language Model Adaptation

For each lecture, the method described in section 3.4 was applied to train a lecture-specific language model (LM) using the vocabulary selected in section 4.1.2, a topic-independent corpora (1280M words) consisting of broadcast news (110M words), parliamentary debates (160M words), printed media (160M words), and web data (850M words), and a lecture-specific corpora (avg. 56M words) consisting of the slides and web documents collected using the method described in section 3.1. The SRILM [11] toolkit was used for LM training and LM interpolation. The resulting lecture-specific LMs obtained a significantly lower perplexity compared to the baseline lecture-independent model as shown in Table 1. On average, the lecture-dependent LMs reduced perplexity by 23.5%.

4.3. Lecture-dependent Speech Recognition

Using the automatically selected vocabularies and lecture-specific language models, we performed speech recognition of each lecture using the automatic speech recognition

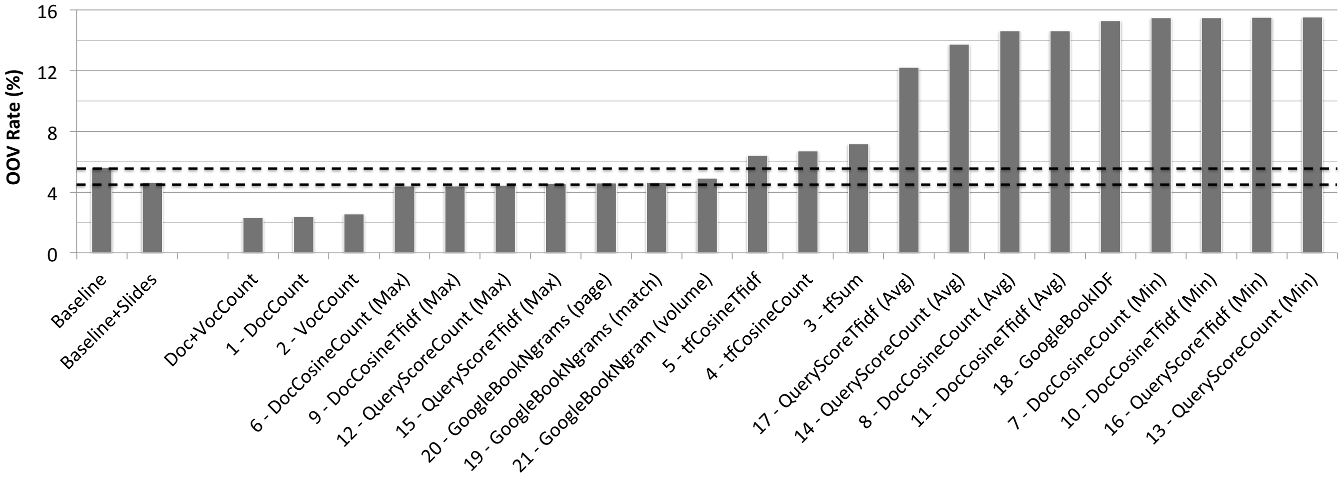

Figure 6: Average OOV rate for all features (40k vocabulary).

	Baseline	Adapt LM
Lecture 1	344.1	261.4 (24.0%)
Lecture 2	352.0	291.3 (17.3%)
Lecture 3	325.0	192.2 (40.9%)
Lecture 4	247.1	207.1 (16.2%)
Lecture 5	274.3	170.1 (38.0%)
Lecture 6	241.3	229.9 (4.8%)
Avg. Improvement	-	**23.5%**

Table 1: Language Model Perplexity (40k Vocabulary)

(ASR) component of our lecture translation system (section 2). Recognition was performed using the Janus speech recognition toolkit using speaker adaptated acoustic models. The German ASR system was trained with around 150 hours of audio data. Speaker adaptive recognition was performed using both feature and model-space adaptation. The acoustic model had 4000 codebooks and each codebook had at most 64 Gaussian mixtures determined by merge-and-split training. Semi-tied covariance and boosted MMI discriminative training was performed during model training. The features for the acoustic model was the standard 39-dimension MFCC. We concatenate adjacent 15 frames and perform LDA to reduce the dimension to 42 for the final feature vectors.

We evaluated the speech recognition accuracy of four different systems. The word-error-rate (WER) results for a 40k vocabulary are shown in Table 2. The lecture-independent baseline system obtained an average WER of 34.3% across the six lectures used in this evaluation. When vocabulary selection (described in section 3.2) was performed using linear feature combination score (Doc+VocCount) an average WER of 32.4% was obtained, a 5.8% relative reduction compared to the baseline system. With LM adaptation (described in section 3.4), an average WER of 33.9% was obtained, a 1.3% relative reduction compared to the baseline system. Applying both, vocabulary and LM adaptation, led to an average WER of 30.1%, a 13.2% relative reduction compared to the baseline system. On average, vocabulary selection obtained higher recognition accuracy than LM adaptation alone, but the biggest gain was obtained by combining both, vocabulary selection and language model adaptation. Although, the improvement was not equally large across all lectures the proposed approach always improved speech recognition accuracy.

Vocabulary Selection		X		X
LM Adaptation			X	X
Lecture 1	43.1	42.4	42.7	41.0 (5.0%)
Lecture 2	34.9	35.7	34.3	34.2 (2.0%)
Lecture 3	33.4	27.3	34.7	26.2 (21.6%)
Lecture 4	28.3	23.9	28.5	22.5 (20.6%)
Lecture 5	28.4	28.8	25.5	21.2 (25.3%)
Lecture 6	37.4	36.4	37.6	35.7 (4.4%)
Average Improvement	-	**5.8%**	**1.3%**	**13.2%**

Table 2: Word Error Rate (40k Vocabulary)

5. Conclusion

Effective adaptation techniques are required to enable lecture transcription and lecture translation systems to perform adequately across the diverse topics that occur in educational and scientific lectures. Our proposed approach solves one of the key issues in current systems, that of selecting an appropriate topic-specific vocabulary for real-time speech recognition. Starting with a seed document, like lecture slides, lecture-related documents are automatically collected from the World Wide Web. Then, a lecture-specific vocabulary is selected using a novel vocabulary selection approach using feature-based ranking scores applied on the collected document corpus. Additionally, the document corpus is used to adapt the language model. Using our approach, the OOV rate was reduced by up to 84.9% (on average by 57.0%) compared to a baseline vocabulary. Furthermore by generating a lecture-specific language model incorporating the retrieved web documents, word error rate was dramatically reduced, obtaining a WER up to 25.3% lower than a lecture-independent Baseline.

6. Acknowledgments

We would like to thank Sebastian Stueker for providing the baseline German speech recognition system used in this work as well as his technical expertise throughout the project. Additionally, we would like to thank the reviewers for their detailed and helpful feedback. This work was supported in part by an interACT student exchange scholarship.

7. References

[1] M. Kolss, M. Wölfel, F. Kraft, J. Niehues, M. Paulik, and A. Waibel, “Simultaneous German-English Lecture Translation,” in *Proc. IWSLT*, 2008, pp. 174–181.

[2] C. Munteanu, G. Penn, and R. Baecker, “Web-based Language Modelling for Automatic Lecture Transcription,” in *Proc. Interspeech*, no. August, 2007, pp. 2353–2356.

[3] H. Yamazaki, K. Iwano, K. Shinoda, S. Furui, and H. Yokota, “Dynamic Language Model Adaptation Using Presentation Slides for Lecture Speech Recognition,” in *Proc. Interspeech*, 2007, pp. 2349–2352.

[4] T. Kawahara, Y. Nemoto, and Y. Akita, “Automatic Lecture Transcription by Exploiting Presentation Slide Information for Language Model Adaptation,” in *Proc. ICASSP*, 2008, pp. 4929–4932.

[5] J. R. Glass, T. J. Hazen, S. Cyphers, I. Malioutov, D. Huynh, and R. Barzilay, “Recent Progress in the MIT Spoken Lecture Processing Project,” in *Proc. Interspeech*, 2007, pp. 2553–2556.

[6] S. Vogel, “Using Noisy Bilingual Data for Statistical Machine Translation,” in *Proc. EACL*, 2003, pp. 175–178.

[7] H. Soltau, F. Metze, C. Fügen, and A. Waibel, “A one-pass decoder based on polymorphic linguistic context assignment,” in *Proc. ASRU*, 2001, pp. 214–217.

[8] S. Vogel, Y. Zhang, F. Huang, A. Tribble, A. Venugopal, B. Zhao, and A. Waibel, “The CMU statistical machine translation system,” in *Proc. MT Summit IX*, vol. 9, 2003, p. 54.

[9] J.-B. Michel, Y. K. Shen, and A. P. Aiden, “Quantitative Analysis of Culture Using Millions of Digitized Books,” *Science*, vol. 331, pp. 176–182, Dec. 2011.

[10] G. Salton and C. Buckley, “Term-Weighting Approaches in Automatic Text Retrieval,” *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.

[11] A. Stolcke, “SRILM-An Extensible Language Modeling Toolkit,” in *Proc. ICSLP*, vol. 2. Citeseer, 2002, pp. 901–904.

[12] S. Chen and J. Goodman, “An empirical study of smoothing techniques for language modeling,” Technical report TR-10-98, Computer Science Group, Harvard University, Tech. Rep., Aug. 1998.

[13] S. Stüker, K. Kilgour, and J. Niehues, “Quaero Speech-to-Text and Text Translation Evaluation Systems,” in *Proc. HLRS*. Springer, 2010, pp. 529–542.