AUTOMATIC KEYWORD EXTRACTION FOR THE MEETING CORPUS USING SUPERVISED APPROACH AND BIGRAM EXPANSION

Fei Liu, Feifan Liu, Yang Liu

Department of Computer Science
The University of Texas at Dallas
{feiliu, ffliu, yangl}@hlt.utdallas.edu

ABSTRACT

In this paper, we tackle the problem of automatic keyword extraction in the meeting domain, a genre significantly different from written text. For the supervised framework, we proposed a rich set of features beyond the typical TFIDF measures, such as sentence salience weight, lexical features, summary sentences, and speaker information. We also evaluate different candidate sampling approaches for better model training and testing. In addition, we introduced a bigram expansion module which aims at extracting "entity bigrams" using Web resources. Using the ICSI meeting corpus, we demonstrate the effectiveness of the features and show that the supervised method and the bigram expansion module outperform the unsupervised TFIDF selection with POS (part-of-speech) filtering. Finally, we show the approaches introduced in this paper perform well on the speech recognition output.

Index Terms— keyword extraction, meeting transcripts, TFIDF, feature selection

1. INTRODUCTION

Keywords can provide important information about the content of documents. However, pre-annotated keywords are often not available for spoken documents, such as meeting transcripts. Recent research has focused on a few meeting understanding tasks (such as summarization, topic segmentation, browsing), but not much on automatic keyword extraction.

There has been various previous work on keyword extraction, primarily on different text domains. TFIDF-based selection has been widely used [1, 2]. It is computationally efficient and performs reasonably well. Keyword extraction has also been treated as a supervised learning problem [1, 2, 3], where a classifier is used to classify candidate words into positive or negative instances using a set of features. Other research for keyword extraction has also taken advantage of semantic resources [4], Web-based metric, such as PMI score (point-wise mutual information) [3], or graph-based algorithms (e.g., [5] that attempted to use a reinforcement approach to do keyword extraction and summarization simultaneously).

Meeting speech is intrinsically different from written text. For example, there are typically multiple participants in the meeting, the discussion is not well organized, the speech is spontaneous and contains disfluencies and ill-formed sentences. Whether existing approaches can be successfully applied to this domain is a question. In this paper, based on the previous keyword extraction work, we propose a supervised approach to automatic extraction of keywords for meeting transcripts. Features that have been found useful for text domain are not necessarily useful or maybe even unavailable for the meeting genre, such as the title or structural information like paragraphs. We utilize a rich set of well-motivated features for this task, such as lexical features to represent whether the sentence is a decision making sentence, the relationship between the keywords and the summary sentences. We perform feature selection to evaluate the effectiveness of various features, as well as sampling to select word candidates. In addition, we introduce a bigram expansion module that uses Google to extract "entity bigrams". Our method significantly outperforms the TFIDF baseline with POS filtering. The same improvement is also observed when using the recognition output.

2. KEYWORD EXTRACTION APPROACHES

Our task is to extract keywords for each of the topic segments in the meeting transcript. Therefore by "document", we mean a topic segment in the following of the paper.

2.1. Supervised Framework

In the supervised approach, a maximum entropy (MaxEnt) classifier is used to determine whether a unigram word is a keyword (binary classification). Each candidate word is represented by a variety of features, explained below.

(A) **Features Used**

- TFIDF. These include: TF, IDF, and TFIDF. The term frequency (TF) for a word w_i in a document is the number of times the word occurs in the document. The inverse document frequency (IDF) value is $log(N/N_i)$, where N_i denotes the number of the documents containing word w_i, and N is the total number of the documents in the collection.
- Position features. These features represent where a word first appears, defined as its position normalized by the total number of words or sentences in the document, referred as 'dis-word' and 'dis-sent' respectively.
- Stopword features. We generate a stopword list by sorting all the words in an increasing order of their IDF values. Three binary features are defined: 'sw-200', 'sw-300', and 'sw-500', to denote whether a candidate word is on the top 200, 300, and 500 of the list. A word with a low IDF means it occurs in many documents and is not topic indicative.
- Sentence features. These are extracted from the sentences containing the word. Feature 'sent-score' is the salience score of a sentence, calculated based on its Cosine similarity to the entire meeting under the vector space model. Feature 'sent-len' is the length of the sentence. If a candidate word appears in several sentences, we empirically use the maximum length and the highest salience score among those sentences.
- Lexical features. These include three feature classes: lex-prp, lex-jj, lex-context. We notice that keywords often appear in

decision making (DM) sentences, similar to the findings in [6]. Instead of building a DM-sentence detection component, we develop a group of lexical features to capture the characteristics of these DM sentences. According to [6], decision making conversations are more likely to contain personal pronouns *we* than *I* and *You*, therefore we use all the personal pronouns that co-occurred with a specific candidate word as a set of lexical features, as well as the total number of personal pronouns, and its normalization by the sentence length. This feature class is represented by 'lex-prp'. Since empirical analysis in [7] showed that subjective words and phrases can influence the importance of sentences, we further include adjective words in the sentences as features (represented by lex-jj). Finally, we also include all the words before and after a candidate word as features (lex-context).

- Summary features. Keyword extraction and summarization tasks have been considered beneficial to each other in text processing [5]. Our statistics on meeting transcripts also confirm that keywords are more likely to occur in the summary compared to the rest of the sentences. Therefore, we extract the following features from summary sentences: a binary feature indicating whether a candidate word has appeared in the summary (summary-in); its frequency in the summary (summary-tf); normalized frequency by the summary length (summary-tf-norm); and the ratio of its occurrence in summary sentences and non-summary sentences (tf-ratio).
- Speaker features. We use these to represent who has said a candidate word. Feature 'spkr-num' is the number of speakers who have said that word. Feature 'spkr-weighted' is the weighted sum of those speakers, with the weight for a speaker calculated using the proportion of sentences from that speaker in the document.

(B) **Candidate Words Resampling**

One way for model training and testing in this supervised approach is to simply use all the words as instances. In this paper, we propose to use a resampling technique to select a subset of the candidate words. This helps address the imbalanced data problem during training, since only a small percent of words are labeled as keywords by human subjects (2.6% in dev set). In addition, during testing, this sampling approach will keep those candidates that are more likely to be keywords and reduce the total number of candidate words.

word candidates	word cov. (%)	KW cov. (%)	P/N Ratio
All words	100	98.77*	0.0268
I: POS sampling	63.85	96.33	0.0415
II: TFIDF	3.88	40.37	0.3793
(II) + keywords in (I)	5.36	96.33	0.9052

Table 1. Words/keywords preserved and the positive to negative ratio using different candidate word sampling. Note that the keyword (KW) coverage is not 100% when using all the words due to some human annotation errors.

We investigated different strategies for resampling. First, we use POS information to filter unlikely keywords. Words labeled with verb, noun and adjective tags are preserved to form a candidate keyword collection. Second, we develop a more aggressive strategy to resample the candidate words by using both POS tags and TFIDF scores as the selection criteria. Candidate words that yield the highest TFIDF scores while satisfying the above POS restrictions are collected for training and testing. Based on the analysis on the development set data (described in Section 3.1), we use 15 candidate words for each document. Table 1 shows the percentage of words/keywords that are preserved and the positive to negative ratio using the development set. Since many reference keywords are not preserved using this TFIDF-based selection, we also consider a third approach that adds the missing keywords that are in the first POS-based selection (last row in the table).

(C) **Generating Unigram Keyword Hypotheses**

We use the posterior probability, $p(C = keyword|features)$, from the MaxEnt classifier to generate keyword hypotheses. Using a default threshold of 0.5, all the words with a posterior probability higher than that are selected as keywords. If there are fewer than 5 keywords generated in this way for a topic, we select the top 5 words with the highest confidence scores from the MaxEnt classifier. This strategy results in at least 5 keywords for each topic (unless there are not that many candidate words). In the development set, we found this yields better results than using exactly 5 keyword hypotheses.

2.2. Bigram Expansion

Looking at the human annotation in the development set, we found that bigrams contribute about 44% of the human extracted keywords. Key phrases containing 3 words or more are much more rare in human annotations. Thus we proposed a method to extract bigram keywords. First we compute the TFIDF scores for all the bigrams in a document. Then similar to the POS constrain used for word resampling, we predefine six POS patterns to filter bigrams, i.e., "nn+nn", "nn+nns", "jj+nn", "jj+nns", "nnp+nn", and "nnp+nnp" (tags used in Penn Treebank tagset). The top N bigrams with the highest TFIDF scores as well as satisfying the POS requirement are selected. N is set to 6 since this setting achieves a good balance of precision and recall.

We propose two ways to refine the bigram selection sequentially in order to further improve precision. First, we use Web resources to recognize those bigrams that are used frequently to represent well-defined entities (e.g., "knowledge engineering", "town hall"). Our aim here is to not select frequent bigrams which cannot represent specific entities (e.g., "other stuff"). For this goal, we create a query to Google that is composed of the 6 bigrams selected based on TFIDF and POS, but reverse the order of the two consisting unigrams in each bigram. The first 10 results returned by Google are examined. If the snippet under each link contains an original bigram candidate (a unigram match is not enough), we take it as an "entity bigram" and hypothesize it as a key phrase. For example, in a meeting transcript, the top ranked 6 bigrams are "computer scientists, system design, re-formulating things, big presentation, system stuff, other linguists", and using Google-based selection, we obtain two candidates: "**computer scientists**" and "**system design**". The idea for this method is that if the words in a bigram are very strong combinations, this bigram can still be found by Google even when we do not preserve their original sequence information in the query and even when it is used along with many other query terms. Our next refinement process utilizes the confidence scores assigned by MaxEnt classifier. For all of the words in the bigram candidates from the previous Google-based selection, it removes those unigram words whose confidence scores are lower than a predefined threshold. Note that this procedure is word-based. Using these two refining processes, we were able to improve the precision rate from 25.6% to 60% when only measuring the words introduced in this module, and on average there are 1.27 more keyword hypotheses for each document on the development set. This suggests our proposed approach allows us to expand the unigram results without bringing in too many false alarms.

We chose not to refine the bigram selection based on their count

in the document collection. When looking at the human annotated keywords, we observe that those "entity bigrams" selected by human annotators may not occur frequently in the entire corpus (partly because of the small data set). For example, 32 out of the 90 selected bigrams occur less than three times. This also posts great challenges to the traditional frequency-based framework; however, our Google refining process leverages the Web information and can effectively address the data sparsity problem.

3. EXPERIMENTS

3.1. Corpus and Experimental Setup

We used the ICSI meeting corpus [8]. All the meetings have been transcribed and annotated with dialogue acts (DA) [9], topics, and extractive summaries [10]. We recruited 3 computer science undergraduate students to annotate keywords for each topic segment, using 27 selected meetings from the ICSI meeting corpus.[1] Up to 5 indicative key words/phrases were annotated for each topic. In total, we have 208 topics annotated with keywords. The average consistency rate is 22.76% and 5.97% among any two and all three annotators, measured using the topic segments excluding the digit recording part and the topics labeled with "Chitchat". This suggests people do not have a high agreement on keyword selection. Therefore, during the evaluation process, we consider both the highest performance with respect to the three annotators and their average scores.

To measure the system performance, we use recall, precision, and F-measure. The comparison between system hypotheses and human annotation is performed on a unigram basis, that is, bigrams will be split into words for performance measurement. We consider a lenient measurement where stemming is used to determine whether two words are matched. For example, if a human annotated keyword is "transcriber", and a system generated keyword is "transcribers", a credit will be given. We used the Porter Stemmer [12].

The speech recognition (ASR) output is obtained from a state-of-the-art SRI conversational telephone speech system [13], with a word error rate of about 38.2% on the entire corpus. DAs and topic boundaries are obtained by aligning human annotated boundaries to the ASR words. We used the TnT POS tagger [14] trained from the Switchboard data to tag the meeting transcripts. The summary sentences are generated by picking the top 5% of the sentences with the highest cosine similarity scores with the entire document. We used 6 meetings as our development set (the same 6 meeting as in [10]) to optimize our keyword extraction methods, and the rest 21 meetings for final testing.

3.2. Experimental Results

3.2.1. Feature Effectiveness

We conducted feature selection using human transcripts on the development set to determine a subset of features that yield the best performance. For this experiment, we use POS-based word selection in our training and testing set. Features are selected based on feature classes described in Section 2.1 (A), not individual features. Instead of using classification accuracy, the average F-score is used as the metric since this is our ultimate performance measure. The selection procedure is shown in Figure 1. We first conduct forward feature selection (FFS) using all the original feature classes. This is done by adding one feature in each iteration that results in the greatest improvement or the least system degradation (if there is no improvement by adding any feature) until all the features are added. A performance curve is plotted for this FFS process. Next we remove the feature that degrades performance the most in the previous FFS step, then perform FFS using this new feature set and re-plot the performance curve. This process is repeated until the performance curve monotonically increases. Finally, we examine all the curves and find the performance peak and the corresponding feature subset. Since TFIDF related features are shown to be effective in many keyword-related tasks, we fix those as the minimum feature set and perform the above feature selection using the rest of the features.

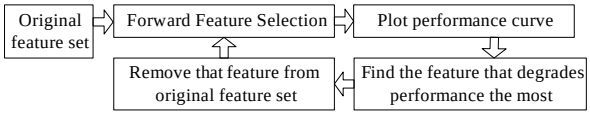

Fig. 1. Flow chart of feature selection.

Table 2 shows the selected features, with their performance when added incrementally based on the order determined in FFS. We found that stopword features, lexical features, and summary-related features are chosen as useful features, and that using a subset of the features can outperform all the features. The model trained using the selected feature subset shows that the top features (with the largest weights in the MaxEnt model) are stopwords for the negative class and TFIDF value for the positive class. This is as what we expected. In another analysis, we also found that the quality of the summaries plays a key role, and that using human annotated summaries outperforms the features extracted from the system generated summaries.

Feature	Feature Description	Avg F (%)
tfidf	TF, IDF, TFIDF	16.33
+ stopword	position in stopword list	20.85 *
+ lex-prp	personal pronouns	23.97 *
+ lex-jj	adjectives	25.03
+ summary-in	whether in system summaries	24.80
+ summary-tf-norm	normalized TF in system summaries	25.08
+ dis-sent	position feature measured by sentence number	23.00
+ tf-ratio	ratio of occurrence between summary sentences and non-summary sentences	25.79 *
All features		23.17

Table 2. Feature selection results. * means that the difference when adding a feature class is statistically significant at $p < 0.05$.

3.2.2. Using Different Candidate Keyword Lists

We examine the effect of word sampling using the feature classes selected above. We trained models on the three candidate word lists as described in Section 2.1 (B). For testing, we can only use POS- or TFIDF-based selection. Results are shown in Table 3. As we can see, using POS-based resampling technique for training generally yields the best performance. The increase of positive to negative ratio in the other two methods, TFIDF-based selection or TFIDF + keywords in POS method, seems to introduce more false alarms and does not improve the models. However, using TFIDF selection to generate test set outperforms the POS-based one, since it only keeps those good candidates and is able to reduce ambiguity during testing for the models. We will use this setting in the following experiments.

3.2.3. Results on Both Human Transcripts and ASR Output

We report the results on human transcripts in Table 4 for the 21 meetings. The baseline is an unsupervised TFIDF approach that selects

[1] We selected these 27 meetings because they have been used in other studies for topic segmentation and summarization [10, 11].

Train	F(%)	Test	
		POS	TFIDF
POS	Max	34.38	35.76
	Avg	25.79	26.68
TFIDF	Max	25.37	35.07
	Avg	19.60	25.95
TFIDF + keywords in POS	Max	13.88	32.35
	Avg	12.96	24.01

Table 3. Results for different candidate word resampling strategies.

the top 5 unigrams (unless there are fewer candidate words) with the highest TFIDF scores while satisfying the POS constrain described in Section 3.2.2. The supervised results are the leave-one-out cross validation results on the 21 meetings. The models are trained using the features selected in Section 3.2.1. When adding bigram expansion to the unigram results, duplicate unigrams were removed. We can see that in general, the supervised framework obtained significantly better results than the TFIDF+POS baseline ($p < 0.05$ using McNemar testing). The bigram expansion process improved performance further, especially the recall rate.

Methods		P(%)	R(%)	F(%)
TFIDF+POS	Max	33.83	28.12	30.71
	Avg	24.63	27.18	25.44
Supervised	Max	37.13	33.36	35.15
	Avg	26.19	32.39	28.44 *
Supervised	Max	36.31	35.76	36.04
+ bigram	Avg	25.52	35.33	29.05 *

Table 4. Results on 21 meetings using human transcripts.

Finally, we investigate the impact of ASR output on our approaches. We re-performed feature selection on the ASR dev set and selected four more features: sent-len, sent-score, dis-word, and lex-context, described in Section 2.1 (A), compared to the features on human transcripts. A re-evaluation of candidate word sampling yielded the same resampling strategies, i.e., POS-based selection for training and TFIDF+POS for testing. Table 5 shows a comparison of F-scores when adopting different extraction methods on ASR output. As can be seen, the performance of both unsupervised and supervised systems degrades compared to using human transcripts. This is not surprising. In fact, we found that only 59.74% of the human annotated keywords appear in ASR output, that is, the upper bound of recall is very low. Moreover, the quality of system-generated summaries and POS tagger on ASR output degrades, which unavoidably affected the features used for keyword extraction. Notice that both the supervised method and the bigram expansion still yield some gain compared to the unsupervised TFIDF+POS baseline, but not as great as on the human transcript condition.

Methods		P(%)	R(%)	F(%)
TFIDF+POS	Max	26.88	22.50	24.50
	Avg	18.84	21.26	19.59
Supervised	Max	26.77	22.66	24.54
	Avg	18.91	22.80	20.23
Supervised	Max	24.97	25.13	25.05
+ bigram	Avg	17.72	25.82	20.55 *

Table 5. Results on 21 meetings using ASR output.

4. CONCLUSION

In this paper, we investigated the problem of automatic keyword extraction in the meeting domain. We adopt a supervised framework and leverage features extracted from meeting specific characteristic such as decision making sentences and system generated summaries. Feature selection and different candidate word resampling techniques prove to be helpful in the supervised method. In addition, we introduced a bigram expansion module which leverages both Web resources and confidence scores from the classifier. Our experimental results on both the human transcripts and ASR output demonstrate that the supervised approach and the bigram module improve keyword extraction performance. Our future work will focus on developing methodologies that are more effective and robust for ASR output, as well as investigating appropriate evaluation metrics that could better deal with low human consistencies.

5. ACKNOWLEDGMENT

The authors thank University of Edinburgh for sharing the annotation of the ICSI meeting corpus, and Shasha Xie for generating the system summaries, ASR alignment, and useful discussions. This research is supported by NSF award IIS-0714132.

6. REFERENCES

[1] E. Frank, G.W. Paynter, I.H. Witten, C. Gutwin, and C.G. Nevill-Manning, "Domain-specific keyphrase extraction," in *Proceedings of IJCAI*, 1999, pp. 688–673.

[2] Y.H. Kerner, Z. Gross, and A. Masa, "Automatic extraction and learning of keyphrases from scientific articles," in *Computational Linguistics and Intelligent Text Processing*, 2005, pp. 657–669.

[3] P. Turney, "Coherent keyphrase extraction via web mining," in *Proceedings of IJCAI*, 2003, pp. 434–439.

[4] L. Plas, V. Pallotta, M. Rajman, and H. Ghorbel, "Automatic keyword extraction from spoken text. a comparison of two lexical resources: the edr and wordnet," in *Proceedings of LREC*, 2004.

[5] X. Wan, J. Yang, and J. Xiao, "Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction," in *Proceedings of ACL*, 2007, pp. 552–559.

[6] P.Y. Hsueh and J. Moore, "What decisions have you made: Automatic decision detection in conversational speech," in *Proceedings of NAACL/HLT*, 2007.

[7] G. Carenini, R.T. Ng, and X. Zhou, "Summarizing emails with conversational cohesion and subjectivity," in *Proceedings of ACL/HLT*, 2008.

[8] A. Janin, D. Baron, J. Edwards, D. Ellis, G. Gelbart, N. Norgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters, "The icsi meeting corpus," in *Proceedings of ICASSP*, 2003.

[9] E. Shriberg, R. Dhillon, S. Bhagat, J. Ang, and H. Carvey, "The icsi meeting recorder dialog act (mrda) corpus," in *Proceedings of SIGdial Workshop*, 2004, pp. 97–100.

[10] G. Murray, S. Renals, J. Carletta, and J. Moore, "Evaluating automatic summaries of meeting recordings," in *Proceedings of ACL 2005 MTSE Workshop*, 2005, pp. 33–40.

[11] M. Galley, K. McKeown, E. Fosler-Lussier, and H. Jing, "Discourse segmentation of multi-party conversation," in *Proceedings of ACL*, 2003.

[12] M.F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, pp. 130–137, 1980.

[13] Q. Zhu, A. Stolcke, B. Chen, and N. Morgan, "Using mlp features in sri's conversational speech recognition system," in *INTERSPEECH*, 2005.

[14] T. Brants, "TnT – a statistical part-of-speech tagger," in *Proceedings of the 6th Applied NLP Conference*, 2000.