Available online at www.sciencedirect.com
ScienceDirect
Computer Speech and Language 24 (2010) 495–514
COMPUTER SPEECH AND LANGUAGE
www.elsevier.com/locate/csl

Improving supervised learning for meeting summarization using sampling and regression

Shasha Xie, Yang Liu *

Department of Computer Science, The University of Texas at Dallas, Richardson 75080, USA

Received 26 September 2008; received in revised form 25 March 2009; accepted 26 April 2009
Available online 15 May 2009

Abstract

Meeting summarization provides a concise and informative summary for the lengthy meetings and is an effective tool for efficient information access. In this paper, we focus on extractive summarization, where salient sentences are selected from the meeting transcripts to form a summary. We adopt a supervised learning approach for this task and use a classifier to determine whether to select a sentence in the summary based on a rich set of features. We address two important problems associated with this supervised classification approach. First we propose different sampling methods to deal with the imbalanced data problem for this task where the summary sentences are the minority class. Second, in order to account for human disagreement for summary annotation, we reframe the extractive summarization task using a regression scheme instead of binary classification. We evaluate our approaches using the ICSI meeting corpus on both the human transcripts and speech recognition output, and show performance improvement using different sampling methods and regression model.

Keywords: Meeting summarization; Imbalanced data; Sampling; Regression

1. Introduction

Automatic summarization is a useful technique to facilitate users to browse a large amount of data and obtain information more efficiently from either text or audio sources. Text summarization has observed significant progress in the past decades, partly due to some benchmark tests such as TIDES, AQUAINT, MUC, and DUC supported and run by DARPA or NIST. Text summarization can be divided into different categories along several different dimensions (Mani and Maybury, 1999). Based on whether or not there is an input query, the generated summary can be query-oriented or generic; based on the number of documents used, summarization can use a single document or multiple documents; in terms of how sentences in the summary are formed, summarization can be conducted using either extraction or abstraction — the former only selects sentences from the original documents, whereas the latter involves natural language generation. Recently sum-

* Corresponding author. Tel.: +1 972 8836618.
E-mail addresses: shasha@hlt.utdallasedu (S. Xie), yangl@hlt.utdallas.edu (Y. Liu).

0885-2308/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.csl.2009.04.007

marization has also been performed in various speech domains, such as broadcast news (Ribeiro and de Matos, 2007; Maskey and Hirschberg, 2006; Maskey and Hirschberg, 2005), lectures (Zhang et al., 2007a,b; Fujii et al., 2007) and meetings (Murray et al., 2005; Buist et al., 2005; Galley, 2006). For any given text or speech input, a good summary is expected to be concise, informative and relevant to the original input.

This paper focuses on extractive summarization using the meeting corpus. Our aim is to select the most representative sentences from a meeting transcript to form a generic summary. There has been increasing interest recently in automatically processing meeting speech, including recognition, summarization, and other understanding tasks in the research community (for example, programs such as AMI/AMIDA, CHIL, and CALO (Hain et al., 2008; Mostefa et al., 2007), and NIST's Rich Transcription evaluation on the meeting domain). Compared to summarization of written text and other speech genres, there are many challenges in the meeting domain because of its more spontaneous style, such as the presence of disfluencies, multiple speakers, less coherence, and often high speech recognition error rate.

Various approaches have been used for text or speech summarization. In this study, we adopt a supervised learning approach for meeting summarization, where for each sentence in the document, a statistical classifier is used to decide whether it is a summary sentence (positive class). This approach has been evaluated for speech summarization using different models, such as hidden markov model (HMM), maximum entropy, conditional random fields (CRF), and support vector machines (SVM; Maskey and Hirschberg, 2006; Buist et al., 2005; Galley, 2006; Zhang and Fung, 2007). Our goal in this paper is to use sampling and regression to improve summarization performance under the classification framework.

Since summary sentences are a small percent of the original documents, there is an imbalanced data problem. In the ICSI meeting corpus (which will be described in Section 3), the average percent of the positive samples is 6.62%. When learning from such imbalanced data sets, the machine learning models tend to produce high predictive accuracy over the majority class, but poor predictive accuracy over the minority class (Maloof, 2003). Different methods have been proposed in the machine learning community for this problem, such as up-sampling and down-sampling, both aiming to make the data more balanced for classifier training. However, this problem has never been studied for the speech summarization task. In this paper, we propose different sampling approaches to deal with the imbalanced data problem by utilizing the original annotated data.

We notice that human annotators often do not agree with others in the selection of summary sentences (Fei Liu and Yang Liu, 2008). In the training set we use, there is only one human annotation available. Because of the large variation in human annotation, a non-summary sentence may be similar to an annotated summary sentence, and other annotators may select this sentence in the summary if multiple annotations were available. We believe these negative samples are noisy and may affect the classifiers to effectively learn to distinguish the two classes. In Table 1, we show two similar sentences in one meeting transcript from the ICSI meeting corpus, where the first sentence was selected by the human annotator to be in the summary. When using binary classification for this task, this kind of labeled data is likely to introduce noise and may be misleading for the classifier.

This problem can be solved by different approaches. The first one is in line with the sampling methods that are motivated to address the imbalanced data problem as mentioned earlier. We reconstruct the training samples to reduce the effect of these confusing instances, either by changing their labels from negative to positive, or removing them from the training set. This sampling method increases the positive to negative ratio of the training set. Changing the labels of instances from negative to positive is an idea similar to up-sampling that increases the number of the positive instances and reduces the number of negative ones; removing the misleading instances from the negative class is a down-sampling method. The difference between our proposed approaches and the traditional sampling methods is that we will focus on the confusable negative examples and thus we expect this will at the same time address the human annotation disagreement problem. The sec-

Table 1
Example of two similar sentences with different labels.

Sentence	Label
I think for word-level this would be ok	+1
For word-level it's alright	−1

ond approach we suggest in this paper is to reframe the summarization task using a regression model instead of binary classification, where we assign each non-summary training sentence a numerical weight according to its similarity to the labeled summary sentences. These weights can provide more elaborate information for the learning models.

This paper centers around these two issues in the statistical classifiers used for meeting summarization, which have not been studied previously. We present empirical evaluations of our approaches using the ICSI meeting corpus. Our results have shown that using different sampling approaches can effectively address the imbalanced data problem and the disagreement of human annotation. Reframing the task using a regression setup instead of classification also helps improve performance. For all of these approaches, we show the improved results for both human transcripts and speech recognition output.

The rest of the paper is organized as follows. Section 2 discusses prior work in summarization and related work on other statistical learning problems. We describe the data used in this paper in Section 3. The summarization approaches we used are introduced in Section 4, including a description of features, the sampling methods, and the regression scheme. The experimental results and analysis are presented in Section 5. Finally, we conclude our paper and discuss some future work in Section 6.

2. Related work

There have been many efforts on summarization, especially text summarization. Many recent studies on automatic text summarization focus on multi-document, multi-model and query-dependent summarization. McKeown et al. (2005) presented an overview of the approaches used in text summarization, and discussed how they can be adapted for speech summarization. Broadcast news speech is the first domain used for speech summarization (Christensen et al., 2003). This domain is similar to the widely used news article domain in many aspects but is different in that it consists of read and spontaneous speech, and there are speech recognition errors (though generally lower than other speech genres), thus presenting a good starting point to evaluate the portability of the classical features and approaches used in text summarization. Broadcast news has continued to receive a lot of attention for speech summarization (Maskey and Hirschberg, 2005; Valenza et al., 1999; Hori et al., 2002). Researchers have also used other genres than broadcast news for speech summarization, such as lecture speech (Zhang et al., 2007a,b; Fujii et al., 2007) and meeting recordings (Murray et al., 2005; Galley, 2006; Penn and Zhu, 2008). In Zhang et al. (2007a), the authors found that the structural features are superior to acoustic or lexical features for broadcast news summarization, but for lecture data, lexical features are more dominant and using only lexical features yielded performance close to using all the features.

Meeting summarization along with other meeting understanding tasks (browsing, detection of action items, topic segmentation, speaker diarization) have recently gained much interest in the research community. There is more data available with various annotation for this domain. Many techniques have been proposed for meeting summarization. Some used unsupervised approaches and relied on textual information only, such as maximum marginal relevance (MMR; Zechner, 2002; Xie and Liu, 2008) and latent semantic analysis (LSA; Murray et al., 2005). Others were based on supervised methods, such as maximum entropy, conditional random fields (CRF; Buist et al., 2005; Galley, 2006), using rich features from textual and acoustic sources. Murray et al. (2005) compared MMR, LSA, and feature-based classification approach, and showed that human judges favor the feature-based approaches. Buist et al. (2005) used a maximum entropy model for meeting summarization based on lexical, structural, speaker and dialog acts features, and showed performance improvement upon a baseline system that selected all the utterances longer than ten words. Galley (2006) proposed using skip-chain CRF to model non-local pragmatic dependencies between paired utterances (e.g., question–answer pairs) that typically appear together in summaries, and showed that these models outperform linear-chain CRFs and Bayesian models. Our prior work evaluated the effectiveness of a variety of features and demonstrated that using a subset of features can outperform using all of the features (Xie et al., 2008).

For the meeting summarization task, how the imbalanced data affects supervised classifiers has never been evaluated in previous studies. We discuss some work related to the imbalanced problem in this section as it has been investigated in some similar speech and language processing tasks. There is no prior work of using con-

tinuous labels and a regression model for summarization, thus we will leave the description of the general regression techniques in Section 4.3 when needed.

The classification performance often degrades when faced with the imbalanced class distributions (Provost, 2000). Most of the classification algorithms are developed to maximize the classification accuracy; however, when the class distribution is imbalanced, the classifier can still achieve a high accuracy even though it fails to detect or classify the minority class (which is often the more important class for most tasks). A common practice for dealing with imbalanced data sets is to rebalance them artificially using "up-sampling" (e.g., replicating instances from the minority class) and "down-sampling" (selecting some samples from the majority class). In addition to modifying the data distribution, it is also possible to modify the classifier (Wang and Japkowicz, 2008). Liu et al. (2006) investigated the use of different sampling approaches for the task of sentence boundary detection in speech. However, the imbalanced data problem has not been evaluated for meeting summarization in most of the feature-based classification approaches, which is a goal of this paper.

3. Corpus and experimental setup

We use the ICSI meeting corpus (Janin et al., 2003), which contains 75 recordings from natural meetings (most are research discussions). Each meeting is about an hour long and has multiple speakers. These meetings have been transcribed, and annotated with dialog acts (DA; Shriberg et al., 2004), topic segmentation, and extractive summaries (Murray et al., 2005). For extractive summary annotation, the annotators were asked to select and link DAs from the transcripts that are related to each of the sentences in the provided abstractive summaries (see Murray et al., 2005 for more information on annotation). Fig. 1 shows a sample from one of the human transcripts, where each line corresponds to a DA, and the ID at the beginning of each line (marked by S*) is the speaker ID. In this excerpt, three sentences (18, 19, and 25) were marked as the summary sentences by the annotator. From this example, we can see that meeting transcripts are significantly different from the input for text summarization (e.g., news article) in that it is very spontaneous, contains disfluencies and incomplete sentences, has low information density, and involves multiple speakers.

The automatic speech recognition (ASR) output for this corpus is obtained from an SRI conversational telephone speech system (Zhu et al., 2005), with a word error rate of about 38.2% on the entire corpus. We

[1] S1 yeah if you breathe under breathe and then you see af go off then you know -pau- it's p- picking up your mouth noise [laugh]
[2] S2 oh that's good
[3] S2 cuz we have a lot of breath noises
[4] S3 yep
[5] S3 test [laugh]
[6] S2 in fact if you listen to just the channels of people not talking it's like [laugh]
[7] S2 it's very disgust-
[8] S3 what
[9] S3 did you see hannibal recently or something
[10] S2 sorry
[11] S2 exactly
[12] S2 it's very disconcerting
[13] S2 ok
[14] S2 so um
[15] S2 i was gonna try to get out of here like in half an hour
[16] S2 um
[17] S2 cuz i really appreciate people coming
***[18] S2 and the main thing that i was gonna ask people to help with today is -pau- to give input on what kinds of database format we should -pau- use in starting to link up things like word transcripts and annotations of word transcripts**
***[19] S2 so anything that transcribers or discourse coders or whatever put in the signal with time-marks for like words and phone boundaries and all the stuff we get out of the forced alignments and the recognizer**
[20] S2 so we have this um
[21] S2 i think a starting point is clearly the the channelized -pau- output of dave gelbart's program
[22] S2 which don brought a copy of
[23] S3 yeah
[24] S3 yeah i'm i'm familiar with that
***[25] S3 i mean we i sort of already have developed an xml format for this sort of stuff**
[26] S2 um
[27] S2 which
[28] S1 can i see it
[29] S3 and so the only question is it the sort of thing that you want to use or not
[30] S3 have you looked at that
[31] S3 i mean i had a web page up
[32] S2 right

Fig. 1. Excerpt of a meeting transcript with summary sentences shown in bold.

align the human transcripts and ASR output, then map the human annotated DA boundaries and topic boundaries to the ASR words, such that we have human annotation for the ASR output. In this paper we use human annotated DA boundaries as sentence information and perform sentence-based extraction.

The same 6 meetings as in Murray et al. (2005) are used as the test set in this study. Furthermore, 6 other meetings were randomly selected from the remaining 69 meetings in the corpus to form a development set, then the rest is used to compose the training set for the supervised learning approach. The development set is used to determine the sampling rates and analyze different weighting methods for sampling and regression. Each of the meetings in the training and development set has only one human annotated summary, whereas for the test meetings, we use three reference summaries from different annotators for evaluation. For summary annotation, human agreement is quite low (Fei Liu and Yang Liu, 2008). The average Kappa coefficient among the three annotators on the test set ranges from 0.211 to 0.345. The lengths of the reference summaries are not fixed and vary across annotators and meetings. The average word compression ratio for the test set is 14.3%, and the mean deviation is 2.9%. These statistics are similar for the training set.

To evaluate summarization performance, we use ROUGE (Lin, 2004), which has been used in previous studies of speech summarization (Zhang et al., 2007b; Murray et al., 2005; Zhu and Penn, 2006). ROUGE compares the system-generated summary with reference summaries (there can be more than one reference summary), and measures different matches, such as N-gram, longest common sequence, and skip bigrams. In this paper, we will use ROUGE F-measures. The options we used in this study are the same as those used in DUC: stemming summaries using Porter stemmer before computing various statistics ($-m$); averaging over the sentence unit ROUGE scores ($-t$ 0); assigning equal importance to precision and recall ($-p$ 0.5); computing statistics in the confidence level of 95% ($-c$ 95) based on sampling points of 1000 in bootstrap resampling ($-r$ 1000).

4. Supervised approach to meeting summarization

The extractive summarization task can be considered as a binary classification problem and solved using supervised learning. In this approach, each training and testing instance (i.e., a sentence) is represented by a set of indicative features, and positive or negative labels are used to indicate whether this sentence is in the summary or not. In this paper, we use support vector machines (SVM) (the LibSVM implementation Chang and Lin, 2001) as the classifier because of its superior performance in many binary classification tasks. During training, an SVM model is trained using the labeled training data. Then for each sentence in the test set, we predict its confidence score of being included into the summary. The summary for the test document is obtained by selecting the sentences with highest scores until the desired compression ratio is reached.

4.1. Features

We extract a variety of features, similar to those in Xie and Liu (2008). In this study, we focus on textual information and do not use acoustic or prosodic features since there is only very limited gain by adding those features in spontaneous speech (Penn and Zhu, 2008). Table 2 lists all the features we use in this study.

As illustrated in the example in Fig. 1, summary sentences tend to be long, an observation similar to text summarization (Kupiec et al., 1995). So we first extract length related features for a sentence, including the sentence length, and the number of words in it after removing stop words. We also include the length information of the previous and the next sentence. Similar to Galley (2006), we use "Unigram" and "Bigram" features, which are the number of frequent words and bigrams in the sentence, computed based on the list we automatically generated (containing words whose frequency is higher than a certain percent of the maximum frequency among all words). Previous work has shown that the first appearing nouns and pronouns in a sentence provide important new information (Maskey and Hirschberg, 2005; Christensen et al., 2004), therefore we use features to represent the number of nouns or pronouns that appear for the first time in a sentence.

Cosine similarity is widely used to calculate the similarity of two text segments. Each document (or a sentence) is represented using a vector space model, and the cosine similarity between two text segments (D_1 and D_2) is

Table 2
List of features in supervised classifier for extractive summarization.

Feature	Feature description
Len I, II, III	Length of previous, current and next sentence
Num I, II, III	Number of words in previous, current and next sentence (after removing stopwords)
Unigram	Number of frequent words
Bigram	Number of frequent bigrams
Noun I, II, III	Number of first appearing nouns in previous, current and next sentence
Pronoun I, II, III	Number of first appearing pronouns in previous, current and next sentence
Cosine	Cosine similarity between the sentence and the whole document
TF I, II, III	Mean, Max, and Sum of TF
IDF I, II, III	Mean, Max, and Sum of IDF
TFIDF I, II, III	Mean, Max, and Sum of TF*IDF
Speaker I, II, III	Main speaker or not for previous, current and next sentence
Same_as_prev	Same as the previous speaker or not
SUIDF I, II, III	Mean, Max, and Sum of SUIDF
ITF I, II, III	Mean, Max, and Sum of ITF
TTFITF I, II, III	Mean, Max, and Sum of TTF*ITF

$$\mathrm{sim}(D_1, D_2) = \frac{\sum_i t_{1i} t_{2i}}{\sqrt{\sum_i t_{1i}^2} \times \sqrt{\sum_i t_{2i}^2}} \tag{1}$$

where t_i is the term weight for a word w_i, for which we use the TF–IDF (term frequency, inverse document frequency) value. The IDF values are calculated using the 69 training meetings. For both the human transcripts and ASR output, we split each of the 69 training meetings into multiple topics based on the topic segment annotation in the corpus, and then use these new "documents" to calculate the IDF values. We also derive various TF and IDF related features (e.g., max, mean, sum) for a sentence following the setup in Galley, 2006; Christensen et al., 2004 that have shown these features are useful.

For each meeting, we find the most talkative speaker (who has said the most words) and speakers whose word count is more than 20% of the most talkative one. These are called main speakers. Each sentence is then labeled with whether it is said by the main speaker, and whether the speaker is the same as the previous one. To capture how term usage varies across speakers in a given meeting, we adopt the feature "SUIDF" introduced in Murray and Renals (2007). The hypothesis for this feature is that more informative words are used with varying frequencies among different meeting participants, and less informative words are used rather consistently by different speakers.

Even though the meeting transcripts are not as organized as broadcast news speech (which generally consists of better story segments), they can still be divided into several parts, each with its own topic. We believe that topic segmentation contains useful information for summarizing a meeting recording, and thus introduce some topic related features to better capture the characteristics of different topics in a meeting. These are based on the so-called topic term frequency (TTF) and inverse topic frequency (ITF). Both of them are calculated on a topic basis for each meeting transcript. The TTF is the term frequency within a topic, and the ITF values are computed as

$$\mathrm{ITF}(w_i) = \log(\mathrm{NT}/\mathrm{NT}_i)$$

where NT_i is the number of topics containing word w_i within a meeting, and NT is the total number of topic segments in this meeting. Note that ITF values are estimated for each meeting, whereas the IDF values are calculated based on the entire corpus. Our hypothesis is that this meeting specific ITF might be more indicative of a specific topic in this meeting.

4.2. Addressing the imbalanced data problem

As we mentioned in Section 1, summary sentences are much fewer than the non-summary sentences for a meeting transcript, thus there is an imbalanced data problem for the summarization task. Our method to deal

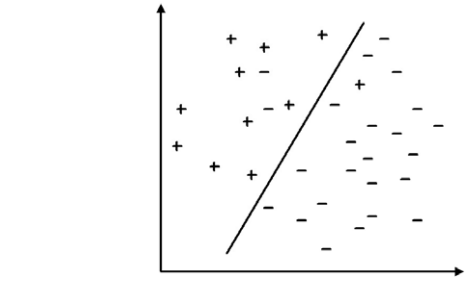
with this problem is to reconstruct the training instances and increase the positive to negative ratio. We propose three sampling approaches in this paper: up-sampling, down-sampling, and re-sampling.

4.2.1. Up-sampling and down-sampling

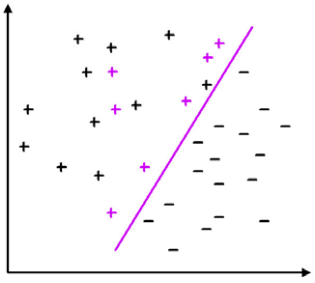
Different from the traditional up-sampling (e.g., replicating positive samples) or down-sampling (e.g., randomly selecting or removing negative samples) methods, the idea of our proposed approaches is to focus on the negative samples which are most similar to the positive ones. For up-sampling, the labels of these confusable samples will be changed from negative to positive. This reduces the number of negative samples, increases the positive samples, and thus increases positive to negative ratio. For down-sampling, these selected negative samples will be removed from the training set.

In Fig. 2, we illustrate how up-sampling and down-sampling change the labels of the instances and the decision boundaries. Fig. 2a shows the original distribution and the hyperplane in SVM for separating the positive and negative instances. After up-sampling, the negative samples that are similar to the positive samples (likely to be close to the decision boundary or in the decision region for the positive class) are changed to positive ones. The hyperplane is moved accordingly, as shown in Fig. 2b. In down-sampling, negative instances close to the decision boundary are removed, also resulting in a change of the decision boundary (Fig. 2c). The new hyperplane based on the new training set after sampling now may be able to better classify instances in the data. For example, the positive instance closest to the boundary in Fig. 2c was labeled as negative by the original classifier, but is now correctly labeled using the new model. We will evaluate the impact of the number of the negative samples that are selected for a label change or removal in these sampling methods. If few samples are selected, the effect of the noisy instances may still be there; if too many instances are selected, additional noise may be introduced because some unimportant sentences are marked as summary sentences.

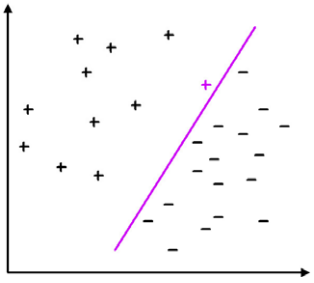
In order to find out these confusable samples, we will assign a weight for each non-summary sentence, which measures its similarity to the reference summary sentences. We then select the sentences with high weights as the confusable samples for further processing. We examine different similarity measures: cosine similarity (as shown in Eq. (1)) and ROUGE score (ROUGE-1 F-measure). For each of the two measures, we use the score of the sentence to the entire reference summary, as well as the maximum and mean value of the sim-



(a) Original distribution



(b) After up-sampling



(c) After down-sampling

Fig. 2. Illustration of up-sampling and down-sampling for binary classification. (a) Original distribution. (b) After up-sampling. (c) After down-sampling.

ilarity scores with individual summary sentences. The two methods, cosine similarity and ROUGE are similar in the sense that they both measure the word match (counting the matched words), but they use different measurement: cosine score normalizes the matches (dot product) by the product of the length of the two vectors; ROUGE score is the harmonic mean of the precision and recall (number of matched words normalized by the length of each individual vector). In addition, for the cosine similarity measure, TFIDF values are used as the term weights, whereas IDF information is not used in ROUGE. Since ROUGE is the final evaluation metric for summarization, we expect that it might be a better similarity measure for sampling. In total, we have 6 different weighting scores, as listed in Table 3. Their impact will be compared in Section 5.2.

Note that the up-sampling and down-sampling methods above also account for the human annotation disagreement to some extent. For example, a negative instance will be relabeled as a positive one if it is similar enough to the positive samples. This is likely to be the case if multiple human annotations were available.

4.2.2. Re-sampling

Both up-sampling and down-sampling methods are used on training set, and the learned models are applied to the test set. In the third sampling method, referred as re-sampling, we perform selection in both training and testing. Fig. 3 shows the flow chart for this method. During training (left part of the figure), each training instance is assigned a weight, and we preserve the samples with higher weights. This process is supposed to give higher weights to positive instances and lower weights to negative ones, and thus the selection procedure preserves most of the positive instances and removes negative instances. This will increase the positive to negative ratio.

Table 3
Weighting measures used in sampling for non-summary sentences.

Weighting methods		Description
Cosine	ALL	Cosine similarity to the entire reference summary
	MAX	Max or mean of cosine similarity to each reference
	MEAN	summary sentence
ROUGE	ALL	ROUGE score based on the reference summary
	MAX	Max or mean of ROUGE scores using each
	MEAN	reference summary sentence

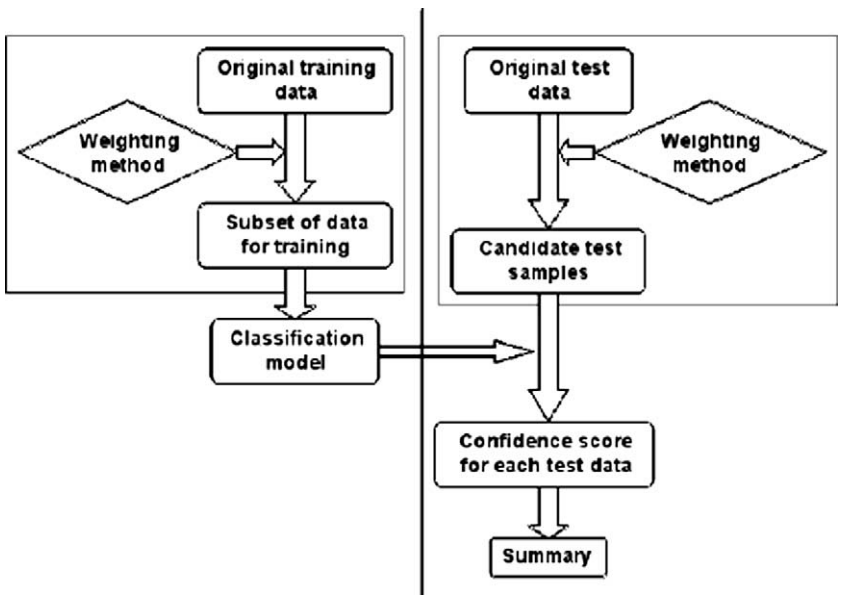


Fig. 3. Flow chart for re-sampling method.

Testing is a two-pass process, as shown in the right part of Fig. 3. The first step does a pre-selection of candidates from the test set that are likely to be positive instances, and then the classifier learned from the training set is applied to these candidate instances to determine summary sentences. We use the same weighting methods as in training to select candidate sentences with high weights in the first pass. Since most of the sentences ignored are non-summary ones, this does not have a negative impact, but rather allows the model to focus on the more likely candidate sentences.

Since a weighting method is needed for both training and testing in re-sampling, we can not use those weighting measures as used for up-sampling and down-sampling, because those are only used during training and they need the reference summary for similarity computation. Therefore we propose two new methods for computing the sentence salience score: one is the sum of the TFIDF values of the words in the sentence; the other is the cosine similarity of the sentence and the entire document. The main difference between these two methods is that cosine similarity is normalized by the sentence length. These two methods can be computed without any information of the labeled summary for a given training or testing transcript. In fact, these two methods are often used in unsupervised extractive summarization to select summary sentences (Xie and Liu, 2008).

Note that this re-sampling is applied to both positive and negative instances in the training set, different from up-sampling and down-sampling that keep all the original positive instances. In terms of the negative samples removed from the training data, this re-sampling method can be thought of as another down-sampling approach. Unlike the down-sampling method that we proposed above in Section 4.2.1, this re-sampling removes instances that are further away from the decision boundary (since they have low similarity scores to the entire document). In order to verify that after removing the sentences with lower weights, the remaining samples still include most of the positive samples, and the training data is more balanced, we calculate the average coverage of the original positive sentences and the percentage of positive instances after re-sampling using different sampling rates for the training data. Results are demonstrated in Fig. 4. We can see that the top 50% sentences can preserve 94.3% of the positive sentences when using TFIDF scores as the selection criteria. The positive percentage after re-sampling is much higher than that in the original data (6.62%). Fig. 4 also shows that TFIDF scores outperform cosine similarity in terms of the coverage of positive samples or the percentage of positive sentences. In general, both of these two weighting methods give higher scores to summary sentences than non-summary ones. We then select a subset of the sentences with higher weights as the instances for training, or the candidates for testing.

4.3. Regression

Another problem using statistical learning for meeting summarization is that it may not be optimal to treat the summarization task as a binary classification problem – two similar sentences may be annotated with two

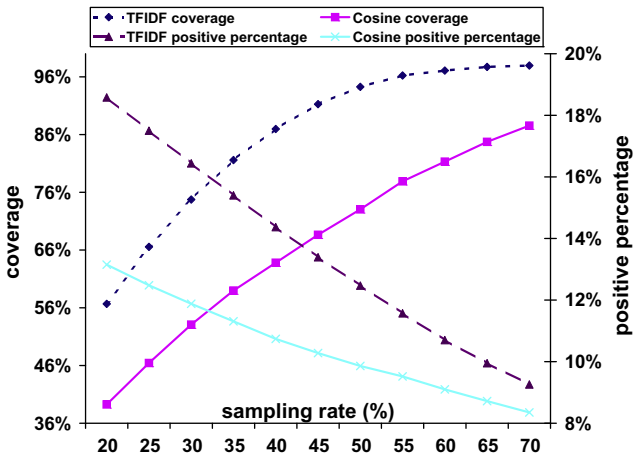


Fig. 4. Coverage of the original positive samples (left *Y*-axis) and the percentage of positive samples in the selected data (right *Y*-axis) using TFIDF and cosine scores as selection criteria for different re-sampling rates.

different labels because of the preference by different annotators or the need to avoid redundancy for summary selection or other reasons, as shown in the example in Table 1. With such kind of confusion, it may be hard for the model to learn to separate summary and non-summary sentences. Therefore, instead of using binary labels, we hypothesize that the summarization task can be modeled as a regression problem, where the labels are numerical numbers representing the importance of the sentences. We expect that the fine-grained weights can provide more information of each sentence's significance and help train a more discriminative model.

The idea of assigning salience weight for the training instances is similar to the weighting methods we used for up-sampling and down-sampling. We keep the positive label (+1) for the summary sentences, and compute target labels for those non-summary sentences using the same 6 weighting measures listed in Table 3. Table 4 shows the labels using Cosine_max for the example sentences used in Table 1. The non-summary sentence has a negative label originally, but now is assigned a high target label because of its similarity to the reference summary sentence. For comparison, we also include the labels for this same example using up-sampling (changed to positive) and down-sampling (this instance is removed).

Once the target labels are assigned to all the training instances, we will use a regression model to learn the underlying function to estimate the target labels. Regression analysis is a statistical tool for investigating relationships between variables. A simple regression model is linear regression that makes the prediction of one variable based on the knowledge of another one when there is a statistically significant correlation between the two variables (Montgomery et al., 2006). Another regression model is logistic regression, which is used for predicting the probability of occurrence of an event by fitting data to a logistic curve (Joanne Peng et al., 2002). This model can be used as a classifier too, where during training, the instances have two classes, and during testing, the model provides a posterior probability of the membership of the test instance. However, our approach here is to use actual target label for each training instances rather than the original binary labels.

The support vector regression (SVR) is the regression model we used in this work (Smola and Olkopf, 1998). Similar to SVM for binary classification, the goal of SVR is to find a function $f(x)$ that has at most ϵ deviation from the actual targets y_i for all the training instances x_i, and at the same time is as flat as possible. Our preliminary experiments showed that SVR outperformed the logistic regression or neural network. During testing, a regression score is predicted for each testing sample, then we use the same method as in the classification approach to select the sentences based on their confidence scores.

5. Experimental results and discussion

5.1. Baseline results

We provide two baseline results for comparison. The first one generates a summary by selecting the longest sentences until reaching the specified length. The second one is the supervised approach that selects the sentences with high confidence scores predicted by the SVM model using all the features we described in Section 4.1. Since the length of the human annotated summary varies for different documents, it is hard to pre-define a proper compression ratio for the summarization system. Moreover, the performance of the system, evaluated by ROUGE scores, is affected by the length of the system-generated summary. Generally, longer summaries have a higher recall rate, but a lower precision score. Therefore we will show results for a few different word compression ratios, measured by the percentage of the words preserved in the summary.

Tables 5 and 6 show the ROUGE-1 (unigram match) and ROUGE-L (longest common sequence match) F-scores for the two baseline systems for the human transcripts and ASR output, respectively. Using sentence

Table 4
Example of new labels using regression, up-sampling, and down-sampling for a non-summary sentence (second row) that is similar to a summary sentence (first row).

Sentence	Original	Regression	Up-sampling	Down-sampling
I think for word-level this would be ok	+1	1.0	+1	+1
For word-level it's alright	−1	0.968	+1	Removed

Table 5
The baseline results (%) for human transcripts.

Compression ratio		13%	14%	15%	16%	17%	18%
Long sent selection	ROUGE-1	52.38	54.50	56.16	57.47	58.58	59.23
	ROUGE-L	51.26	53.35	55.13	56.55	57.69	58.28
Supervised classifier	ROUGE-1	67.25	67.80	67.76	67.56	67.22	66.86
	ROUGE-L	66.39	66.97	67.01	66.80	66.47	66.13

Table 6
The baseline results (%) for ASR outputs.

Compression ratio		13%	14%	15%	16%	17%	18%
Long sent selection	ROUGE-1	62.13	63.11	64.01	64.72	64.65	64.89
	ROUGE-L	60.95	61.96	62.86	63.60	63.59	63.79
Supervised classifier	ROUGE-1	62.83	63.93	64.42	64.73	64.77	64.45
	ROUGE-L	62.15	63.23	63.72	64.02	64.05	63.67

length to select summary yields worse performance than using the classification approach on the human transcript condition; however, the two systems achieve similar results on the ASR output. This finding is consistent with the results reported in Penn and Zhu (2008). Comparing the performance on the human transcripts and ASR output using the classification approach, we see that the results are consistently better for human transcripts on different compression ratios, which is expected. The two ROUGE scores show similar trends for different test conditions. The compression ratio has an impact on summarization performance, and we notice that a higher compression ratio tends to yield better performance for the ASR conditions. Overall, the baseline results are very competitive with the previous work (Murray et al., 2005; Galley, 2006; Penn and Zhu, 2008). For the following experiments, we use the results achieved by using the SVM classifier with all the features as the baseline, and evaluate the performance improvement using our proposed approaches.

5.2. Results for addressing imbalanced data problem

Since our focus here is on the approaches to deal with the imbalanced data problem, we fix the word compression ratio and evaluate the effect of different sampling rates and weighting methods on summarization performance using ROUGE-1 F-measure scores on the development set. We use 14% and 17% word compression ratio for the human transcript and ASR output respectively. These compression ratios are chosen based on the baseline results above.

5.2.1. Experimental results using up-sampling and down-sampling

5.2.1.1. Experimental results for up-sampling. Up-sampling selects negative samples that are similar to the summary sentences and moves them to the positive class. We described 6 weighting methods to measure similarity in Section 4.2.1, three based on cosine similarity, and the other three using ROUGE scores.

Fig. 5 shows the ROUGE-1 F-scores of up-sampling using the 6 weighting methods and different up-sampling rates on the human transcript for the development set. The X-axis, the sampling rate, is the rate of the current positive samples to the original positive instances. When it is 1, none of the negative instances is changed to positive and there are no newly added positive samples, that is, the results are the same as the baseline system. We can see that different similarity measures and up-sampling ratios have great influence on the system performance. Of all the weighting methods, the best results are obtained using *ROUGE_mean* and increasing the positive samples to 1.5 times of the original number. This yields an F-score of 69.24%, compared to the baseline result of 67.80%. When further increasing the up-sampling rate using similarity measure *ROUGE_mean*, there is a performance drop. For other similarity measures, the trend is not clear – the results are more random and there is more fluctuation.

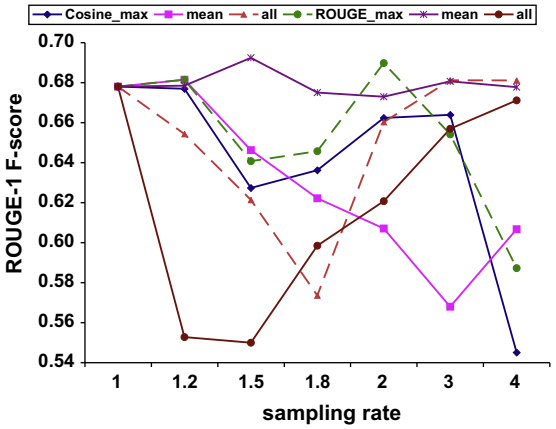


Fig. 5. The up-sampling results on human transcripts.

The results for the ASR condition are shown in Fig. 6. The same setting, 1.5 up-sampling rate and *ROUGE_mean*, also outperforms the baseline result (65.66% vs. 64.77%). However, using the weighting measure *Cosine_mean* and a sampling rate of 4, achieves slightly better result, 65.97%. But the pattern using different sampling rates for the *Cosine_mean* similarity measure is not as clear as for *ROUGE_mean*. Comparing ASR and human transcripts, it seems that a higher up-sampling rate is often preferred for ASR output and there is more fluctuation in the human transcript condition when varying the sampling rates.

Next for a comparison, we show results for the commonly used up-sampling method that replicates the samples in the minority class. The ROUGE-1 F-measure results using different sampling rates are shown in Table 7 for both human transcripts and ASR output. The sampling rate in this experiment is the number of the times that we replicate the summary sentences. When the up-sampling rate is 1, it is the same as the baseline setup. Note that the results shown are mostly for integer sampling rates, where we replicate all of the positive instances. We include one fractional sampling rate, 1.5, since that is the best configuration obtained from our proposed up-sampling method.

For human transcripts, replicating the positive class degraded performance; in contrast, our proposed up-sampling method can yield performance gain. When using ASR output, for some up-sampling rates, there is an improvement, even though we also observe significant performance drop for some up-sampling rates. The best result is similar to that in our proposed up-sampling method (e.g., using *ROUGE_mean* weighting method). Compared to our approach above, the performance variance when changing the up-sampling rates seems to be greater when up-sampling is achieved by replicating minority samples. Overall, for all the up-sam-

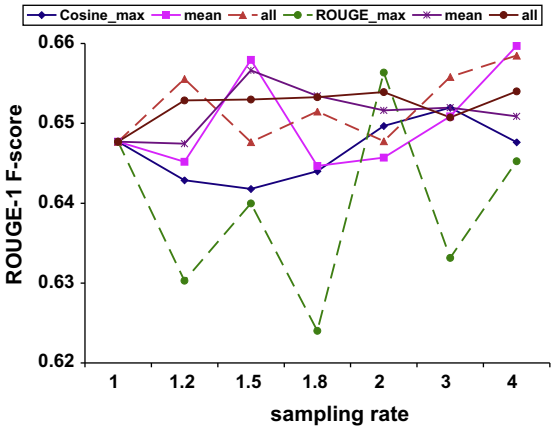


Fig. 6. The up-sampling results on ASR output.

Table 7
The ROUGE-1 F-measure results (%) of up-sampling by replicating the positive samples on both human transcripts and ASR outputs.

Up-sampling rate	1	1.5	2	3	4	5	6
Human	67.80	66.49	63.73	57.39	55.18	57.89	58.36
ASR	64.77	65.38	59.94	66.18	65.57	65.49	66.04

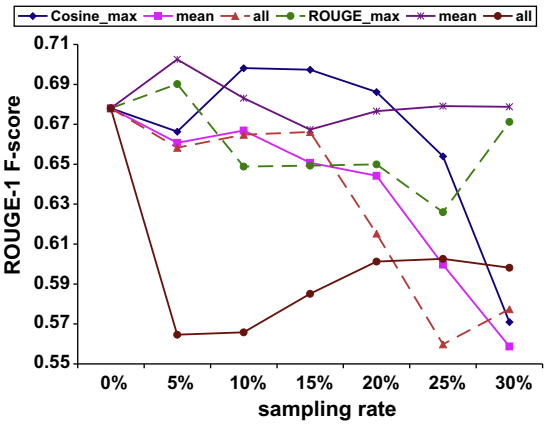


Fig. 7. The down-sampling results on human transcripts.

pling approaches, there is not a consistent correlation between the system performance and how balanced the resulting data set is after up-sampling.

5.2.1.2. Experimental results for down-sampling. The same 6 weighting methods are used for down-sampling, with the goal of removing negative instances with high similarity scores to the summary sentences. Fig. 7 shows the down-sampling results on the development set using human transcripts. The *X*-axis, the down-sampling rate, represents the percentage of the removed negative instances. When it is 0, it means that no negative instances are removed, which is the baseline setup. We see from the results that the best performance is obtained using *ROUGE_mean* as the weighting method with a down-sampling rate of 5%. The performance is improved from 67.80% to 70.28%. The experimental results on the ASR output are shown in Fig. 8. Using the same weighting method and down-sampling rate, we obtain the best score for ASR outputs, 66.42% compared to the baseline score of 64.77%. For *ROUGE_mean* similarity measure, we observe that the best result is achieved with a sampling rate of 5%, then the performance starts degrading when removing more instances from the data set. This observation is consistent for the human transcripts and ASR output.

For a comparison with the down-sampling approach above, we also evaluate a commonly used down-sampling method, i.e., by randomly removing the negative samples from the data set. Because the removed samples are randomly selected, we performed three random sampling runs and obtained the average results from them. Table 8 shows the ROUGE-1 results for human transcripts and ASR output respectively when varying the down-sampling rates. Using the human transcripts, there is only marginal change of the results for different sampling rates. The results on ASR output fluctuate a bit for different sampling rates. There is a performance gain for a few different setups. Our proposed approach outperforms this commonly used down-sampling method for both human transcripts and ASR condition. In addition, our approach has similar patterns on the human transcripts and ASR output, whereas this down-sampling by random selection has a different trend for these two conditions.

5.2.1.3. Discussion. In Table 9, we list the setups (weighting method and sampling rate) which yield the best results on the development set for up-sampling and down-sampling, along with their ROUGE-1 F-measure scores. For most of the experiments, *ROUGE_mean* outperforms the other weighting measures, except for up-sampling on ASR output, where *Cosine_mean* is slightly better. This is consistent with our expectation that

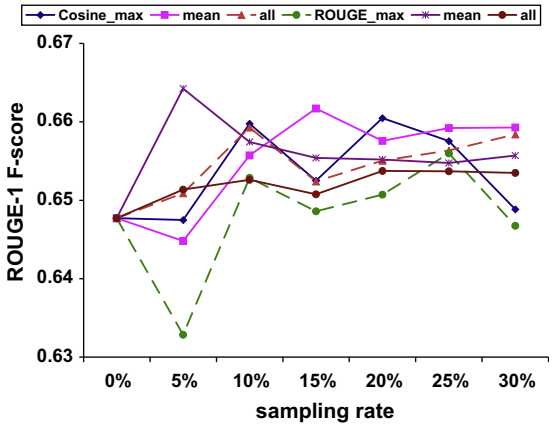


Fig. 8. The down-sampling results on ASR outputs.

Table 8
Results of down-sampling by randomly removing negative samples on both human transcripts and ASR output.

Down-sampling rate (%)	5	10	15	20	25	30
Human	68.53	67.86	67.86	68.34	68.33	68.00
ASR	64.74	65.30	64.31	65.35	66.10	64.52

Table 9
The setup yielding the best results for up-sampling and down-sampling.

		Weighting method	Sampling rate	ROUGE-1
Up-sampling	Human	*ROUGE_mean*	1.5	69.24
	ASR	*Cosine_mean*	4	65.97
Down-sampling	Human	*ROUGE_mean*	5%	70.28
	ASR	*ROUGE_mean*	5%	66.42

ROUGE is the final evaluation metric, and is expected to be a better weighting method for capturing sentence similarity. The best scores are obtained by the mean value of the weighting methods, which suggests that the selected sentences should be the most similar ones to the entire summary, not to a specific sentence (as is done using the max value). In addition, the fact that the average of the cosine similarity or ROUGE scores yields better performance indicates that it is better to give equal weight to different summary sentences than more weight to longer sentences.

5.2.2. Experimental results using re-sampling

For re-sampling, we use different selection criteria to retain some training and test instances. We compare two measurements: TFIDF and cosine similarity, as introduced in Section 4.2.2. Fig. 9 shows the re-sampling results on human transcripts for the development set. The *X*-axis, re-sampling rate, is the percentage of the instances preserved. When it is 100%, we do not delete any instances, so the results are the same as the baseline. We notice that when using cosine similarity scores to keep the top 20% instances, we obtain the best score of 70.41%. The sampling rate is relatively low (20%), which results in an average positive coverage of 39.3% on the training set. This implies that the positive coverage is not the only criteria to evaluate a weighting method for re-sampling. The system performance is also dependent on which sentences are actually preserved by this weighting method, for both positive and negative classes. The experimental results on ASR output are shown in Fig. 10. We observe very different patterns between using ASR and human transcripts. The best score on ASR output is obtained using TFIDF as the selection metric, and keeping the top 35% samples. This yields a performance improvement from 64.77% to 66.27%.