STATISTICAL SENTENCE EXTRACTION FOR INFORMATION DISTILLATION

Dilek Hakkani-Tür

International Computer Science Institute (ICSI)
Berkeley, CA 94704
dilek@icsi.berkeley.edu

Gokhan Tur

SRI International,
Menlo Park, CA, 94025
gokhan@speech.sri.com

ABSTRACT

Information distillation aims to extract the most useful pieces of information related to a given query from massive, possibly multilingual, audio and textual document sources. One critical component in a distillation engine is detecting sentences to be extracted from each relevant document. In this paper, we present a statistical sentence extraction approach for distillation. Basically, we frame this task as a classification problem, where each candidate sentence in documents is classified as relevant to the query or not. These documents may be in textual or audio format and in a number of languages. For audio documents, we use both manual and automatic transcriptions, for non-English documents, we use automatic translations. In this work, we use AdaBoost, a discriminative classification method with both lexical and semantic features. The results indicate 11%-13% relative improvement over a baseline keyword-spotting-based approach. We also show the robustness of our method on the audio subset of the document sources using manual and automatic transcriptions.

Index Terms— information distillation, information extraction, language understanding, speech processing, natural language processing

1. INTRODUCTION

As the amount of available information grows tremendously, methods for directly accessing the relevant information efficiently and effectively have become increasingly important. Now the need is developing methods to extract only the requested information. In the framework of the DARPA GALE program, this process is called *distillation*. For example, given a set of multilingual audio and text sources, the purpose of distillation is to extract the biography of a person, or list arrests from a given organization during a specific time period with explanation. The participants are given a set of query templates with a variable portion. The goal of a distillation system is to output ordered segments called *snippets* that can be considered as an answer to these queries. A snippet can range from a fragment of a sentence to a paragraph. Below is an example query (in which the location and date range are variables) with some related snippets:

Query: *Describe attacks in [the Gaza Strip] giving location (as specific as possible), date, and number of dead and injured. Provide information since [28 Sept 2000].*

Snippets:

- *attack against a school bus filled with Israeli children*
- *There were 45 students and 2 teachers in the bus*
- *The militant Islamic Jihad claimed responsibility*

One critical component for distillation is detecting sentences to be extracted from each relevant document. The user typically is not interested in reading the whole news story but instead just the portion with the requested information content. This process is relevant but not exactly equivalent to document summarization [1], information retrieval [2], question answering [2], or information extraction [3, 4]. In a summarization system, there is usually no predefined query; simply, documents are summarized. Yet, sentence extraction may be part of a summarization system, such as [5, 6, 7]. However, the goal in a summarization system is to extract the most informative sentences that summarize the whole document, so features like the number of named entities in the sentence or rank of the sentence in the document are extremely useful. Recently, Document Understanding Conferences (DUCs) also incorporated question-focused (or query-relevant) summarization in their evaluations [8], where the similarity of each sentence with the question can also be used as a feature. However, the questions in the DUCs do not have a predefined structure such as the query templates.

Question answering systems, on the other hand, act upon a requested question, such as *what is the longest river in the world?* or *list all the European union countries.* In that sense question answering output is very formatted compared to distillation. Distillation usually requires details for the event in question. Even for a query very close to question answering, such as *What is the relationship of Chirac to France*, the answer is not a single word or sentence (such as *He is the president*); instead, anything relating *Chirac* to *France* in the corpora needs to be in the answer set. Information extraction, more specifically event extraction, can in our opinion be considered as the closest match for distillation. Some of the ACE events [4] (such as *arrest* or *attack*) are among the GALE distillation queries. Furthermore recent TREC 2006 evaluations include template-based information retrieval tasks which are similar to distillation as well [2].

In this paper, we focus on sentence extraction for information distillation. The general SRI team approach for information distillation is the work of a bigger team, and will be explained in detail in another paper. In summary, we are given all the data sources to be searched during distillation. The data includes both textual and audio data in multiple languages, namely English, Chinese, and Arabic. We use automatic translations of the non-English data. For audio data we use both manual and automatic transcriptions. The University of Massachusetts INDRI search engine [9] indexes all the data. Then, during runtime when a query is given, the INDRI search engine retrieves candidate documents, considering the dates, the sources of documents to be searched, and so on, as specified in the query. Then the sentence extraction process tries to identify the potential snippets. Finally, similar sentences are clustered into groups. Due to the diversity of the data sources and the noise introduced via speech recognition (ASR) and machine translation (MT), it is important to have a robust method. In this study, we use AdaBoost classification algorithm with lexical (word n-grams) and se-

1-4244-0728-1/07/$20.00 ©2007 IEEE

mantic features to extract sentences relevant to each query.

In the next section we present our approach. Then we provide experimental results comparing our approach with a simple keyword-spotting-based method first for all the data and then using only audio documents.

2. SENTENCE EXTRACTION

The goal of sentence extraction is to tag each sentence as relevant or not given a set of documents relevant to a distillation query. During this study, we focus on sentence extraction for two types of queries used for the DARPA GALE Program:

Query Template 15: *Identify persons arrested from [organization] in [location] and give their name and role in organization and time and location of arrest.*
and

Query Template 16: *Describe attacks in [location] giving location (as specific as possible), date, and number of dead and injured.*

The date and the sources of information that need to be searched are also specified in these queries. The first step is retrieving documents that are relevant to the given query. Then the sentences that are related to the query are extracted. We assume that sentence boundaries are already extracted by either the NYU ACE System [10] for text or the ICSI+ sentence segmentation system for speech input [11]. While a similar information-retrieval-based approach may be used for the sentence extraction, we propose to employ state-of-the-art discriminative classification methods. This is also important for improving the robustness of the system to the noise introduced by the automatic speech recognition and machine translation systems for multilingual and/or audio documents.

2.1. Keyword-Spotting-Based Approach

We built a baseline keyword-spotting-based system in order to compare our results. In this baseline system, each sentence gets a vote if certain keywords (such as *arrest* or *detain* for template 15 and *bomb* or *kill* for template 16) and named entities (such as organization, location, or date) mentioned in the query appear in the sentence. Then the recall/precision curves are drawn according to the number of votes. This can be considered as a grammar-based approach for distillation.

2.2. Classification-Based Approach

Note that the keyword-spotting-based approach has certain drawbacks. It requires human expertise for in-domain knowledge and involvement and it is not robust to changes in the tasks or domains, and noise introduced by ASR and/or MT outputs. Hence, we employ a data-driven (or learning) method for sentence extraction in information distillation.

To train sentence extraction models, we extract negative and positive examples from the given answer keys, which have the relevant snippets and the corresponding document identifiers for each query. The answer keys very rarely include more than one sentence as a response, in which case we split the snippets into sentences. Since the relevant sentences in those answer keys also include the document identifiers, we extract all sentences in those documents as examples, and mark the sentences whose portion are in the answer key as positive examples, and all the rest as negative examples. Here, note that the same sentence can be marked as a positive example by one query and as a negative example by another. When answers are from non-English sources, we use the automatic translation of those answers as positive examples. We believe that this will improve the robustness of the system to the noise introduced by ASR and MT. For the experiments with ASR output we align the automatic hypotheses with reference sentences and extract their class (positive or negative) from the answer keys.

Given the collection of all snippets, $x_1, ..., x_m \in X_j$, from each document $d_j \in D$, where D is the set of documents that have at least one snippet as an answer to a query, we form the set:

$S = \{(x_1, 1), ..., (x_m, 1), (y_1, 0), ..., (y_n, 0)\}$
where y_k is the k^{th} irrelevant sentence in the document d_j, assuming that this document has n irrelevant sentences. Then the sentence extraction for distillation task is defined as estimating the conditional probability, $p(c|s_i)$, $c \in \{0, 1\}$, that sentence s_i is relevant ($c = 1$) or not ($c = 0$). We then return sentences which have $p(c|s_i) > th_q$, where the threshold th_q is estimated using a heldout set for each query q.

During classification we use lexical and semantic features. Lexical features consist of word n-grams obtained from the training examples. Using all word n-grams instead of several keywords as features is also expected to improve the robustness of the system. This can be considered as a query-specific information extraction system, which is supposed to perform better than a generic one. We then augment these features with semantic ones by tagging the raw sentences to mark the instances of organization, location, or dates in the query. Sometimes equivalent terms are also given in the query. For example, the equivalent term for the organization *al-Qaeda* is *al-Qaida*. Similar mapping is done for those phrases as well. Then the classifier is given the word and/or tag n-grams extracted from both the raw sentence and the tagged sentence.

We performed our tests using the Boostexter classification tool [12], an implementation of the Boosting family of classifiers. Boosting is an iterative procedure; on each iteration a weak classifier is trained on a weighted training set, and at the end, the weak classifiers are combined into a single, combined classifier. More formally, output the final classifier for an example x for the class l is defined as:

$$f(x, l) = \sum_{t=1}^{T} \alpha_t h_t(x, l)$$

where $h_t(x, l)$ is the score given by the weak classifier h_t learned at iteration t. $alpha_t$ is the weight of each classifier, typically determined according to the accuracy of that weak classifier. Each weak classifier (e.g., “decision stump”) checks the absence or presence of a feature. Boosting is shown to be a very effective written and spoken language classification tool [12, 13]. Note that our approach is independent of the specific classification algorithm used.

3. EXPERIMENTS AND RESULTS

In this study, we used 46 sample queries, 25 from query template 16 and 21 from query template 15 provided by DARPA GALE project. Detailed data characteristics are shown in Table 1. Since it is beyond the scope of this paper, we assumed that the IR engine returns all the relevant documents and nothing else. The corpus is a collection of Arabic, English, and Mandarin broadcast news, broadcast conversations, newswire, and other written news forms such as blogs, and so on. It mainly consists of TDT-4 and TDT-5 corpora [14] in addition to some recently dated material.

Since the whole distillation task is now framed as a classification problem, we have evaluated the performance of our system using the standard recall and precision metrics and their harmonic mean, F-measure. To obtain these values, we use macro-averaging over

	Template 15	Template 16
Number of queries	25	21
# Snippets (Positive Sents.)	1,346	4,136
# Negative Sents.	5,240	11,138
Number of Documents	515	1793
Avg. Snippets per Document	2.6	2.3
Avg. Snippet Length	19.4 words	18.6 words

Table 1. Data characteristics used in the experiments.

	Template 15	Template 16
Keyword-Spotting	45.42%	52.35%
Lexical Only	48.57%	57.91%
Lexical+Semantic	51.41%	58.20%

Table 2. Macro-averaged F-measures for keyword-spotting-based and statistical approaches.

queries since the number of relevant snippets per query varies significantly.

We have performed n-fold cross-validation where in each fold one query was used for testing, one for tuning the iteration count and decision threshold th_q for Boosting, and the rest for training. The classifier confidence scores are then used to compute the F-measure values at various thresholds.

Table 2 presents our summarized results comparing the keyword-spotting-based approach with the statistical approach. For the keyword-spotting-based approach we chose the best threshold on the test set, an unfair advantage for this approach; for the statistical approach the development set was used to choose the threshold for each query. We also tried using contextual features, such as the rank of the sentence in a document, but results did not change significantly. For both queries, the statistical approach outperformed the keyword-spotting-based approach significantly.[1] The improvement is about 6% absolute in F-measure for both cases. One interesting observation is that, while adding semantic features helps for query template 15, its effect is insignificant for template 16.

Figures 1 and 2 present the aggregate recall/precision curves for the keyword-spotting-based and classifier-based approaches. That is, at each point of the curve, the same threshold is used for all queries. As seen, classifier-based approach generally outperforms the keyword-spotting-based one throughout the curves. An interesting observation is that, for query template 16, the keyword-spotting-based approach performs comparable to classifier-based one in a high precision / low recall region, while this is not the case for template 15.

When we looked at the error distributions, we see that keyword-spotting-based approach is performing poorly for extracting sentences which contain relevant but additional information about the queries. That is, it makes errors when the sentence to be extracted does not have enough number of indicative keywords or phrases. For example this approach misses the second snippet of the example query above, which complements the first one. On the other hand, classification-based approach sometimes miss relevant sentences although they contain the keywords the other approach is looking at. This is due to the discriminative classification approach we are employing, that is, other words in the sentence also matter.

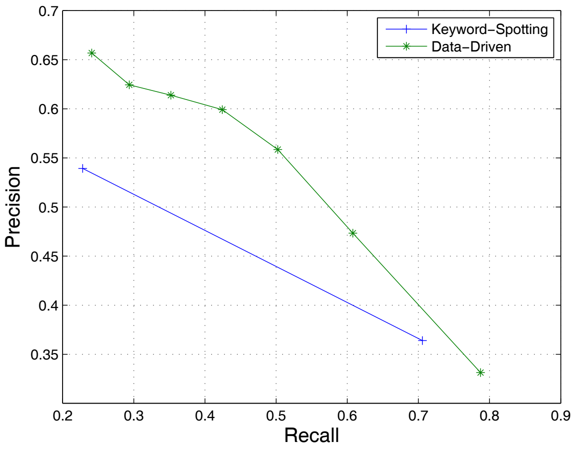

Fig. 1. Recall-Precision curves when using the keyword-spotting-based and classifier-based approaches for query template 15.

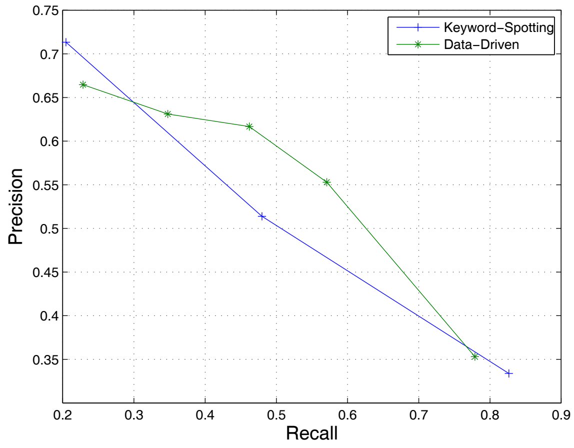

Fig. 2. Recall-Precision curves when using the keyword-spotting-based and classifier-based approaches for query template 16.

As a final experiment we excluded the sentences that appear with both positive and negative classes ambiguously due to the contribution of different queries. This resulted in exclusion of 1-2% of the training data (in total there are 111 such examples for query template 15 and 147 for query template 16). Neither the exclusion nor disambiguation of them (marking them consistently as positive or negative) changed the classification performance significantly.

3.1. ASR Experiments

We then selected the subset of queries which have correct snippets in English audio documents. This gave us 8 queries (6 from template 16, 2 from template 15) with 179 snippets (3.3% of all snippets) in 77 documents in total. Including the negative sentences, we ended up with 837 sentences, 3.8% of all the sentences. Although the sample size is not large enough, we checked the performance of the classifier-based and keyword-spotting-based approaches using both manual transcriptions and ASR output.

[1] according to the Z-test with 0.95 confidence interval

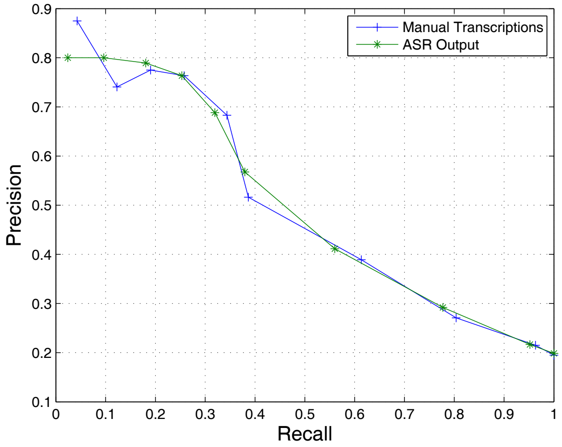

Fig. 3. Recall-Precision curves when using manual and automatic transcriptions.

Figure 3 presents the recall and precision curves for this set. As seen, there is no significant difference in the performance, indicating the robustness of our method. Although the word error rate for this subset of documents is found to be 21%, the data-driven approach is not affected significantly. Note that manual transcriptions are *quick* transcriptions and also include some noise, such as *atck* instead of *attack* and the result on ASR output for such cases was better when they were correctly recognized.

4. CONCLUSIONS

This paper presents a data-driven approach for sentence extraction in information distillation. To the best of our knowledge this is the first study on this topic. We have successfully employed a classification-based approach for this task, improving the task accuracy by around 11%-13% relative.

In this study we are not using any other semantic features, such as other named entities, resolved coreferences, and ACE-style event types. For example, in order to tag a location or organization we require the full name to appear in text, ignoring the coreferences (such as *the terrorist organization*). Similarly, we have the semantic events for the whole corpora, marked automatically by the NYU ACE System, and both *attack* and *arrest* are events used in ACE. Currently, we do not exploit this information either, and plan to investigate them in the future.

We were also given *irrelevant* documents for each of the queries. This is very useful information especially when discriminative classifiers are used. We plan to perform these experiments in the future work. As other future work, we plan to augment our feature set with more semantic features such as named entities, resolved coreferences, and ACE-style event types.

One problem with this task is that it is sometimes not clear whether a sentence must be extracted since it is a subjective decision. So instead of making a binary decision, the data may be divided into more classes such as *very relevant*, *marginally relevant*, and so on.

5. ACKNOWLEDGMENTS

We thank SRI GALE Distillation team members at Columbia University, University of Massachusetts, NYU, and Fair Isaac. This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-06-C-0023. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

6. REFERENCES

[1] "Document understanding conference (DUC)," http://www-nlpir.nist.gov/projects/duc.

[2] "Text retrieval conference (TREC)," http://trec.nist.gov.

[3] *Proceedings of the 7^{th} Message Understanding Conference (MUC-7)*, Fairfax, VA, April 1998.

[4] "Automatic content extraction (ACE)," http://projects.ldc.upenn.edu/ace.

[5] J. Kupiec, J. Pedersen, and F. Chen, "A trainable document summarizer," in *Proceedings of the SIGIR*, Seattle, WA, August 1995.

[6] J. Goldstein, M. Kantrowitz, V. Mittal, and J. Carbonell, "Summarizing text documents: Sentence selection and evaluation metrics," in *Proceedings of the SIGIR*, Berkeley, CA, August 1999.

[7] S. Maskey and J. Hirschberg, "Comparing lexical, acoustic/prosodic, structural and discourse features for speech summarization," in *Proceedings of the Interspeech*, Lisbon, Portugal, September 2005.

[8] H. T. Dang, "Overview of DUC 2005," in *In Proceedings of the DUC*, Vancouver, Canada, 2005.

[9] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft, "Indri: A language-model based search engine for complex queries," in *Proceedings of the International Conference on Intelligent Analysis*, McLean, VA, May 2005.

[10] R. Grishman, D. Westbrook, and A. Meyers, "NYU's English ACE 2005 System Description," Tech. Rep. 05-019, NYU Proteus Project, 2005.

[11] M. Zimmerman, D. Hakkani-Tür, J. Fung, N. Mirghafori, L. Gottlieb, E. Shriberg, and Y. Liu, "The ICSI+ multilingual sentence segmentation system," in *Proceedings of the ICSLP*, Pittsburg, PA, 2006.

[12] R. E. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," *Machine Learning*, vol. 39, no. 2/3, pp. 135–168, 2000.

[13] N. Gupta, G. Tur, D. Hakkani-Tür, S. Bangalore, G. Riccardi, and M. Rahim, "The AT&T spoken language understanding system," *IEEE Transactions on Speech and Audio Processing*, vol. 14, no. 1, pp. 213–222, 2006.

[14] Charles L. Wayne, "Topic Detection and Tracking (TDT) Overview and Perspective," in *Proceedings of the DARPA Broadcast News Tracsription and Understanding Workshop*, Lansdowne, VA, June 1998.