INTEGRATING SEVERAL ANNOTATION LAYERS FOR STATISTICAL INFORMATION DISTILLATION

Michael Levit[1], *Dilek Hakkani-Tür*[1], *Gokhan Tur*[2], *Daniel Gillick*[1]

[1]International Computer Science Institute, Berkeley, CA 94704
[2] SRI International, Menlo Park, CA 94025

ABSTRACT

We present a sentence extraction algorithm for Information Distillation, a task where for a given templated query, relevant passages must be extracted from massive audio and textual document sources. For each sentence of the relevant documents (that are assumed to be known from the upstream stages) we employ statistical classification methods to estimate the extent of its relevance to the query, whereby two aspects of relevance are taken into account: the template (type) of the query and its slots (free-text descriptions of names, organizations, topic, events and so on, around which templates are centered). The idiosyncrasy of the presented method is in the choice of features used for classification. We extract our features from *charts*, compilations of elements from various annotation levels, such as word transcriptions, syntactic and semantic parses, and Information Extraction annotations. In our experiments we show that this integrated approach outperforms a purely lexical baseline by as much as 30% relative in terms of F-measure. We also investigate the algorithm's behavior under noisy conditions, by comparing its performance on ASR output and on corresponding manual transcriptions.

Index Terms— Information Distillation, Question Answering, Statistical Natural Language Processing, Machine Learning

1. INTRODUCTION

Automatic Question Answering is a growing research field in the Speech and Language Processing area. In recent years, the field has experienced rapidly expanding datasets whose sources are no longer restricted to written English but include recognized speech and automatic translations. Besides, the questions themselves have become more elaborate. All these changes suggest that robust, trainable statistical mechanisms should augment (if not entirely replace) pattern-based models.

In Question Answering, one is asked to find answers to queries from a possibly very large collection of documents. We consider the Template-based Question Answering task (termed "Information Distillation" in the DARPA-funded GALE program) where a finite number of query templates are agreed upon in advance and have variable slots that are to be filled with free-text descriptions at runtime. For instance, the template *"Describe the prosecution of* [PERSON] *for* [CRIME]*"* has two slots that could be filled with PERSON=*Saddam Hussein*, CRIME=*crimes against humanity*. One other example is the open-domain template *"Describe the facts about* [EVENT]*"* with possible definitions of the slot EVENT like *"civil unrest in France"* or *"bird flu outbreak in China"*. The output of the Information Distillation system is a list of snippets (sentences or phrases) relevant to the query.

In [1] we described our first explorations of statistical classification for information distillation in the framework that was using the University of Massachusetts INDRI search engine [2] to find relevant documents in response to a query. We extracted n-grams of words and names from the sentences of the retrieved documents and used them as classification features. The most relevant sentences were selected and redundancy was removed. The present paper extends this approach in several significant ways.

First, our classification features are comprised of not only word transcriptions and names, but also of other representations associated with the sentences. Among those we count syntactic parses, semantic predicate-argument structures, and various elements of Information Extraction (IE) annotations. By incorporating all these representations into one coherent structure, we are able to evaluate combinations of elements from different annotation levels and leverage classification results.

Second, we acknowledge the importance of having a separate mechanism for instantiation of query slots in the sentences. In our system, slots are accounted for in two ways: they can either be instantiated directly in sentence text, or reduced to "important information" (meaning) that will be then searched for in the sentences.

We will show that an advanced system extended in this way is not constrained to a particular range of templates, but rather is capable of handling a wide variety of queries provided that training material is available.

The remainder of the paper is organized as follows: We start by reviewing related work on Question Answering systems built on several annotation layers. Next, we present our classification approach describing the nature of classification features we use to decide on a sentence's relevance for a query. We then describe our experimental setup and show results on the GALE Distillation task. Future work and conclusion sections complete the paper.

2. RELATED WORK

The important role Information Extraction (IE) plays in Question Answering has been noted and the advantages of using IE repeatedly demonstrated in the literature [3, 4, 5]. In [6], the authors introduce several layers of IE (that are very similar to the ACE elements and predicate-argument structures that we are using) to constrain the scope of candidate sentences to those containing IE elements associated with a query. However, the associations are revealed using handwritten rules, and combinations of different elements are not evaluated.

More recent publications prefer IE elements from ACE annotation guidelines that are defined by the annual NIST ACE evaluations [7] and include, among other tasks, mention coreference resolution (e.g., pronominal coreference), entity extraction (for people, organizations, locations, etc.), and entity relation and event extraction (e.g., *ownership* relation and *attack* event). For instance, in [4], the presence of relevant ACE events and ACE entities in the vicinity of a sentence is taken as an indicator of the sentence's relevance.

In [3], semantics expressed as predicate-argument structures (*propositions*) were used (together with IE and other annotations) as one of the sources for classification features to answer biographical questions. [8] shows how these propositions, because of their absolute lexical coverage, can be employed to deal with open-domain templates such as *"Describe the facts about* [EVENT]*"*. In particular, propositions extracted from slot formulations are organized into *proposition trees* and instantiated in proposition trees extracted from sentences. [9] demonstrated how syntax and semantics can be incorporated into one coherent structure (syntactic and semantic graph, SSG) to serve the calltype classification task.

3. METHOD

The traditional approach for Information Distillation proceeds as follows:

1. *Information Retrieval (IR) stage:* start by retrieving documents that are likely to contain answers.
2. *Snippet Extraction stage:* use the retrieved documents to select sentences (or parts thereof) that contain answers.
3. *Answer Formulation stage:* combine (rank, remove redundancy and possibly modify) the extracted sentences to form an output.

Here, we are mostly concerned with the second stage and cast it as a classification problem. Given a query and a particular sentence, the task is to classify the sentence as either relevant or irrelevant for the query. The main challenge here is a mindful selection of classification features that, on one hand, must be sensitive enough to pay attention to subtle formulation differences of query slots and, on the other hand, general enough to facilitate classification with respect to previously unseen queries. Applied to the Distillation task as formulated in the GALE Distillation Guidelines [10], these features must respond to query templates and to their slots.

Figure 1 shows an overview of the features our system extracts along with the extraction methods. The next sections explain this process in more detail.

3.1. Classification Features from Charts

Our system uses several categories of features. First, we will describe the features obtained from the so-called *charts*, an amalgam of elements from various annotation levels represented as directed acyclic graphs that we create for each sentence.

A chart of a sentence with N words consists of N+1 states that are connected by arcs labeled with *chart entries*. Chart entries can be of several types. The simplest ones are words, so the trivial chart is just a linear graph whose arcs are words (in their natural order). Other annotation levels can be integrated in charts as well. For the experiments presented in this paper, we have considered the four chart entry types listed below. They can also be seen in Figure 2 that contains the full chart for sentence: *"John gave Mary his car"*, where arc prefixes indicate chart entry type:

- **words** (prefix "w+")
- **part-of-speech tags and syntactic parses** (prefix "s+")
- arguments and targets of PropBank **predicate-argument structures** [11] (prefix "p+")
- **IE elements** including entities, relation and event arguments, but also Timex2 mentions (prefix "a+")

Charts are similar to Syntactic and Semantic Graphs (SSGs) in [9], but incorporate more information and, because of the integrated IE coreference, allow for more flexibility in modeling sentences. Next, we explain n*-gram* and *inclusion* features extracted from them charts.

3.1.1. n-gram Features

Since chart entries of all types are treated the same way, n-gram extracted from charts can consist of entries that are quite heterogeneous in nature. For instance, from the chart in Figure 2, the following n-grams can be extracted (we use "⊎" as a connector in n-gram representations):

a+PER ⊎ *p+give_targ* ⊎ *w+Mary*
w+gave ⊎ *a+PER* ⊎ *s+NP*
w+John ⊎ *s+VP*
a+PER ⊎ *s+VP*

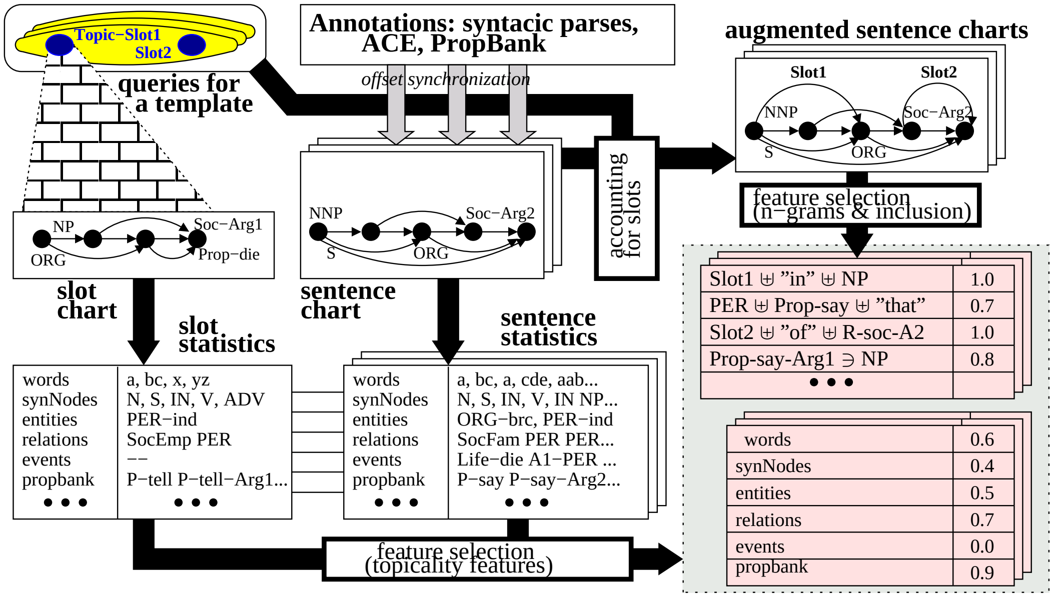

Fig. 1. Extraction of Classification Features from Sentences.

This generation process is very powerful; paired with a task-dependent feature selection process, it allows to keep only features that strike just the right balance between specificity and generality, while ignoring everything else. For instance, if the question is about John's actions, then the third feature in the list above seems most relevant. If we are asked about people's actions in general, then the fourth feature will probably suit our needs better. Finally, if the focus is on the process of giving, then the first feature in the list should be favored with respect to the others.

Even though we use statistical methods to perform feature selection, it is certainly worth mentioning that writing patterns for manual feature selection can straightforwardly be accomplished with the above format. One can in fact use the automatically generated pool of potential pattern candidates to start manual selection.

3.1.2. Inclusion Features

While n-grams contain information about sequences of chart entries, the inclusion features focus on parallelism. They will tell us, for example, that the word *John* is at the same time also syntactic node NP, and an entity of type PER-IND (person, individual), but also the first argument in a relation and arg0 (agent) for the "giving" predicate. These features are obtained by looking at pairs of chart entries that "contain" each other (in terms of the words they account for). Currently, to make the search feasible, we eliminate pairs in which the including chart entry covers significantly fewer words than the included one. In the chart in Figure 2, some of the inclusion features will be ("∋" stands for "includes")

a+Ownership_arg1 ∋ *a+PER-IND*
s+S ∋ *s+VP*
a+VEHICLE-LAND ∋ *w+his*

3.2. Accounting for Simple Slots

In Information Distillation and other Question Answering tasks, we do not know beforehand that the query will be about John, so it is impossible to predict in advance that the feature *w+John* ⊎ *s+VP* will be useful. What the templates do tell us is that the query will be about some person defined in a query slot. This leads us to the idea of creating a special chart entry named SLOT as a generalized place-holder for specific names, organizations etc. that the query is asking about. Note that this is the first time that our chart generation (and feature extraction) algorithm makes use of the actual queries, and the entire procedure up until this point could be done offline.

To find instantiations of slots in a sentence, we do bidirectional weighted bag-of-words pattern matching. The algorithm possesses some linguistic knowledge (like verb nominalizations and synonyms), and some world knowledge (e.g., gazetteer resources). Contributions of individual words to the overall score depend on their parts of speech and occurrence frequencies.

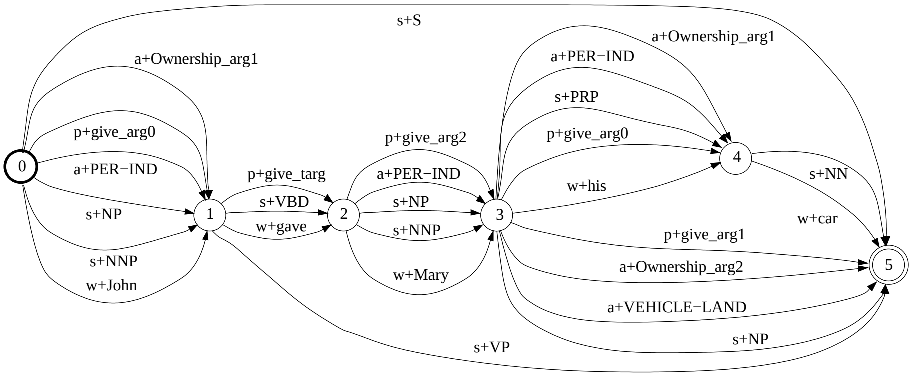

Fig. 2. Chart for sentence *"John gave Mary his car"*; it contains 5 words (prefix w+), 10 syntactic nodes (prefix s+), 4 mentions of 3 entities (prefix a+), arguments of one *ownership* relation, and finally target and arguments of verb *give* (prefix p+).

3.3. Topicality Features

Chart entry SLOT can be used for some query slots, but it is not directly applicable to others. In particular, it seems fairly futile to try instantiating such slot types as topic or event, because of their high expressive variability. For instance, for event formulation *"attacks in India"*, the sentence *"there was an explosion in Delhi"* is perfectly on-topic even though it does not contain a single content word from the original event formulation.

In addition, the nature of chart entry SLOT is quite restrictive, in the sense that *"Al Amin Khalifa Fhimah"*, for instance, would not be recognized as an instantiation of a person-slot formulation *"Al Fhimah"* without further relaxation steps. While this might be a desirable behavior for chart entry SLOT, generally, we would like to see some indicators that a certain similarity is present.

The TOPICALITY features fill these niches. They look at charts computed for slot-spans of the queries and try to instantiate chart entries found there in the chart entries found in the sentences. The instantiation is done separately for several categories of chart entities (which will be described below) and performed for all slots independently.

For each slot/category combination, we consider all relevant entries in the slot chart and try to instantiate them in the sentence chart. Then we average the best achieved instantiation scores to be the score of the topicality feature for this combination. The following topicality categories are supported:

1. words
2. ACE entities of types PER, ORG and GPE+LOC (each one separately and combined)
3. same for ACE relations
4. same for ACE events
5. targets of semantic predicates

For instance, if there are two PER entities in the text of a slot and one of them is instantiated in the sentence chart with a perfect score of 1.0, and another is not instantiated in it at all, the score of the topicality feature TOPICALITY-PER for this slot is set to be $(1.0 + 0.0)/2 = 0.5$.

3.3.1. Taking Advantage of Context Information

While deciding if a sentence is relevant or not, it might be useful to look at its direct neighbors as well. We do this while computing topicality features by augmenting topicality features computed for the sentence in question by topicality features computed for the surrounding sentences.

3.4. Handcrafted Elimination Rules

In [5] we recognized the advantage of restrictive rules involving ACE elements. Here we extend the approach to prune away obvious false positives even before classification takes place. We devised a number of rules involving elements from various annotation levels that must be satisfied in order for a sentence to be considered for classification. For example, for the prosecution template in Section 1 we require that one of the following holds: a) at least one of the topicality features fires on the sentence for the person slot, and crime is directly instantiated as slot, b) person is instantiated as slot, and there is a justice ACE event in the sentence, c) topicality fires for both crime and person, and there is a justice event. The actual rules used in our system considered not only the sentences in question, but also their direct contexts.

4. EXPERIMENTS AND RESULTS

We now describe the experimental setup in which our system was tested.

4.1. Queries

We have used our system to find answers to queries from several templates as defined by GALE Y2 guidelines [10]. Tem-

#	text
1	List facts about event: [EVENT]
8	Describe the prosecution of [PERSON] for [CRIME]
12	Provide a biography of [PERSON]
15	Identify persons arrested from [ORGANIZATION] and give their name and role in the organization and time and location of arrest
16	Describe attacks in [LOCATION] giving location, date and number of dead and injured

Table 1. Query templates considered for system evaluation

plate numbers and their formulations are given in Table 1. For each of the templates in this table, several training and test queries have been provided along with *answer keys*: labels "relevant" and "irrelevant" given to each sentence by human labelers. All answer keys were generated for documents that contained at least one relevant sentence, according to the labelers.

4.2. Corpus

Documents that supplied answer keys for training and test queries came from different corpora. We used the English text part of the GALE Y1 training data for training and the English text part of GALE Y2 corpus for our main test. Overall, there was an average of 24 training and 9 test queries per template. About 2700/500 considered documents contained 38K/16K sentences, and supplied an average of 72/39 relevant snippets per query for the training/test corpus. Not only did the test corpus have a significantly smaller proportion of relevant snippets (11% as opposed to 21% in the training corpus), but its queries were judged as subjectively more difficult by human observers. In addition, to investigate performance of our algorithm on ASR data, we created a smaller test corpus of ASR transcriptions, that will be explained in Section 4.4.

All documents were processed in advance to obtain syntactic parses and PropBank annotations using Charniak and ASSERT parsers [12]. We used NYU's toolkit Jet [13] to produce ACE-annotations. Similar processing was applied to all queries, and in particular to their slot spans.

4.3. Experiments: Advantage of the Integrated Approach

The aim of our first set of experiments was to investigate the potential of using other annotation layers in addition to words. We created parallel sets of classification features extracted from sentence charts that contained only words (baseline), as well as features from "mixed" charts with all annotation layers introduced above. The first and second rows of Table 2 show the F-measures for class "relevant" achieved for each template in both experimental setups with an SVM-classifier.

template	1	8	12	15	16	overall
words	0.42	0.42	0.25	0.40	0.46	0.39
all features	0.47	0.51	0.43	0.42	0.52	0.47
all w. rules	0.50	0.63	0.44	0.46	0.56	0.51
accept-all	0.40	0.32	0.17	0.22	0.33	0.29

Table 2. SVM classification results (F-measure) for a) word bigrams, b) all classification features, c) all features and handwritten rules, d) accept-all trivial classifier.

We see that for all templates, introducing additional annotation levels to produce classification features and using topicality features resulted in very significant improvements. Other evidence of success is that even using trigrams instead of bigrams in the words-only baseline could not narrow the gap. It is worth mentioning however that, given the baseline's ignorance of the slots, its performance looks remarkably good. What happens is that the system assumes that the selected documents *are* about the slots, so that it only needs to be looking for words that indicate the right template. For instance, if an arrest is mentioned in the document, it is presumably an arrest of the slot-person. Another reason is most clearly articulated by the open-domain template 1. Here, the seemingly high F-measure for the word-only case is not much better than the one of a trivial classifier that marks all sentences as relevant (last row of Table 2). In fact, unlike other templates, entire selected documents are often dedicated to the event in question, so that most of the sentences in them are on-topic (e.g., for event *"Civil unrest in France"* this proportion is about 65%).

We are planning to present a detailed discussion about individual contributions of ACE and PropBank annotations as part of n-grams and inclusion features in a separate paper. At this time, however, we would like to point out that using rules from Section 3.4 that contain those annotations to eliminate obviously irrelevant sentences appeared to be very helpful. The third row of Table 2 convincingly demonstrates this fact.

4.4. Performance on Speech Transcriptions

Our final series of experiments investigated performance degradation on noisy data, such as ASR output. For that, we used six queries from template 16 for which we had 842 snippets (20% of them relevant) from 77 documents annotated by human labelers. The models were retrained on English text documents after eliminating training material for these six queries. We then compared distillation results for ASR output (WER≈15%) and corresponding manual transcriptions.

Table 3 shows that there is a performance degradation when using ASR output instead of noise-free annotations. However, the F-measure loss is not prohibitive for using the models. We also see that the advantage of using IE elements and other feature sources diminishes as we switch to ASR out-

	transcriptions	ASR output
words	0.57	0.48
all features	0.65	0.49

Table 3. Comparison of distillation F-measure between ASR output and corresponding manual transcriptions for template 16; word-only and all-features cases considered.

put (3% relative as opposed to 14%). We believe that this is caused by the fact that all the systems we used for syntax, semantics, and IE had been trained on text, and thus have difficulty dealing with noise.

5. FUTURE WORK

The classification framework we presented in this paper offers many opportunities for feature analysis and improvement. Our next goal is to examine individual contributions of various feature types to sentence relevance classification. We also plan to aid classification by taking advantage of natural word similarity and introducing word clusters as features. Furthermore, we hope to enhance semantic features by adding NomBank [14] annotations to the ones of PropBank. At the same time, further exploration of high coverage semantic parses [8, 15] could be a viable alternative. Yet another promising direction seems to come from extracting n-grams from syntactic or dependency trees. This, however, must be done cautiously as parsing quality drops when switching to ASR data.

6. CONCLUSION

We presented a system for Information Distillation that extracts relevant sentences for templated queries with variable slots. Our strategy was to combine sentence annotations from several levels (such as word transcriptions, syntactic and semantic parses, and IE elements) into time-synchronous directed acyclic graphs (*charts*) and extract classification features from these charts to learn sentence relevance. By using the system for GALE Distillation, we improved the average F-measure for five templates from 0.39 (with words only) to 0.51. Furthermore, we have demonstrated that the system can be used on noisy ASR data as well.

7. 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.

8. REFERENCES

[1] D. Hakkani-Tür and G. Tur, “Statistical Sentence Extraction for Information Distillation,” in *International Conference on Acoustics, Speech, and Signal Processing*, Honolulu, HI, April 2007.

[2] 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.

[3] R. Weischedel, J. Xu, and Licuanan A., “A Hybrid Approach to Answering Biographical Questions,” in *New Directions in Question Answering*, M. Maybury, Ed., chapter 5, pp. 59–69. MIT Press, 2004.

[4] B. Schiffman, K. McKeown, R. Grishman, and J. Allan, “Question Answering Using Integrated Information Retrieval and Information Extraction,” in *HLT/NAACL*, Rochester, April 2007.

[5] D. Hakkani-Tür, G. Tur, and M. Levit, “Exploiting Information Extraction Annotations for Document Retrieval in Distillation Tasks,” in *InterSpeech*, August 2007.

[6] R. Srihari and W. Li, “Information Extraction Supported Question Answering,” in *Eighth Text Retrieval Conference (TREC-8)*, Gaithersburg, MD, November 1999.

[7] LDC, “ACE English Annotation Guidelines for Entities,” Tech. Rep., 2005.

[8] M. Levit, E. Boschee, and M. Freedman, “Selecting On-Topic Sentences from Natural Language Corpora,” in *InterSpeech*, August 2007.

[9] D. Hakkani-Tür, G. Tur, and A. Chotimongkol, “Using Syntactic and Semantic Graphs for Call Classification,” in *Workshop on Feature Engineering for Machine Learning in Natural Language Processing at ACL’05*, Ann Arbor, MI, June 2005.

[10] BAE, “Go/No-Go Formal Distillation Evaluation Plan for GALE,” Tech. Rep., 2006.

[11] P Kingsbury, M. Palmer, and M. Marcus, “Adding Semantic Annotation to the Penn TreeBank,” in *Human Language Technology Conference (HLT)*, San Diego, CA, 2002.

[12] S. Pradhan, W. Ward, K. Hacioglu, J. Martin, and D. Jurafsky, “Shallow Semantic Parsing using Support Vector Machines,” in *HLT/NAACL*, Boston, MA, May 2004.

[13] R. Grishman, D. Westbrook, and A. Meyers, “NYU’s English ACE 2005 System Description,” Tech. Rep. 05-019, NYU, 2005.

[14] A. Meyers, R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young, and R. Grishman, “The NomBank Project: An Interim Report,” in *HLT-NAACL Workshop: Frontiers in Corpus Annotation*, A. Meyers, Ed., Boston, MA, May 2004, pp. 24–31.

[15] M.-C. de Marneffe, B. MacCartney, and C. D. Manning, “Generating Typed Dependency Parses from Phrase Structure Parses,” in *LREC*, Genoa, Italy, 2006.