INTERSPEECH 2009 BRIGHTON

Classification-Based Strategies for Combining Multiple 5-W Question Answering Systems

Sibel Yaman[1], *Dilek Hakkani-Tur*[1], *Gokhan Tur*[2], *Ralph Grishman*[3], *Mary Harper*[4], *Kathleen R. McKeown*[5], *Adam Meyers*[3], *Kartavya Sharma*[5]

[1] International Computer Science Institute, [2] SRI International
[3] Computer Science Department, New York University
[4] Hopkins HLT Center of Excellence, University of Maryland
[5] Computer Science Department, Columbia University

{sibel,dilek}@icsi.berkeley.edu,gokhan@speech.sri.com,{grishman,meyers}@cs.nyu.edu
mharper@umd.edu, kathy@cs.columbia.edu,ks2580@columbia.edu

Abstract

We describe and analyze inference strategies for combining outputs from multiple question answering systems each of which was developed independently. Specifically, we address the DARPA-funded GALE information distillation Year 3 task of finding answers to the 5-Wh questions (*who, what, when, where,* and *why*) for each given sentence. The approach we take revolves around determining the best system using discriminative learning. In particular, we train support vector machines with a set of novel features that encode systems' capabilities of returning as many correct answers as possible. We analyze two combination strategies: one combines multiple systems at the granularity of sentences, and the other at the granularity of individual fields. Our experimental results indicate that the proposed features and combination strategies were able to improve the overall performance by 22% to 36% relative to a random selection, 16% to 35% relative to a majority voting scheme, and 15% to 23% relative to the best individual system.

Index Terms: Question answering, Systems for spoken language understanding

1. Introduction

Information distillation aims to analyze and interpret massive amounts of multilingual speech and text archives and provide queried information to the user. In general, a distillation system first processes the given user query using an information retrieval (IR) system to find the relevant documents among huge document collections. Each sentence in these documents is then processed to determine whether or not it answers the user's query.

The goal in the third year of the DARPA-funded GALE information distillation task is to predict answers to the 5-W questions (*who, what, when, where,* and *why*) for the top-level predicate, i.e., the main predicate, for each and every given sentence. More specifically, the answer to *who*, the actor/agent, should be the logical subject of the sentence, which is indicated with a "by clause" in passive sentences. The answer to *what* should contain the main predicate plus its logical object. The answer to *when*, the temporal argument, should include specific times ('3 days ago'), non-exact times ('prior to his term'), and adverbs of frequency/duration (e.g. 'for a year', 'sometimes', 'often'). The answer to *where*, the locative argument, refers to a physical location ('behind the building') or a metaphorical location ('in the speech', 'in the process'). Finally, the answer to *why*, the causative argument, should be an explicitly triggered expression for reason/cause ('as a result of the crisis', 'because of the explosion').

The 5-W extraction task is important in many respects. First, it eliminates the errors that originate from the IR component of a typical distillation system. Second, since there is only one sentence for which each and every 5-W question is answered, it is possible to evaluate the success of a system in finding answers to specific questions from the same sentence. Third, it also becomes possible to evaluate the premise of the state-of-the-art techniques for the isolated task of extracting answers, because, for the task described, it is critical that a sophisticated syntactic, semantic, and contextual processing of documents and queries is performed.

While document and information retrieval in response to user queries has been well studied, finding exact answers to a question is less well-studied. Answering factoid questions (i.e., questions like *What is the capital of France?*) using web or huge data collections typically makes use of the redundancy of information [1]. In addition to the work on written texts, many systems have been developed that can operate with either spoken queries or spoken document collections [2]. The combination strategies developed in this paper are similar to those developed for combining multiple systems for semantic role labeling [3], which is the task of finding the arguments of each of the predicates in a sentence. In summary, the nodes of a tree are to be marked with semantic role labels (SRLs) such as *ARG0* to denote a *subject*. Combining various SRL systems is different in that involves a heavy analysis of parse trees.

In this paper, we first describe three systems to extract answers to 5-W questions, and then we move our focus to the problem of combining these systems. We designed a feature extraction module that makes use of novel features. Using the so-formed feature vectors, we developed two combination strategies to predict the system that would return as many correct answers as possible. One primary contribution is the analysis we performed to answer the specific questions of (i) what kind of features provide useful information for this task, (ii) what kind of a combination strategy works best, and (iii) in which question and in which genre are the combination strategies most effective. Our experimental results indicate a significant improvement over any individual system, as well as combination strategies that are not based on learning.

2. Answering 5-W Questions

The GALE information distillation task specifies finding answers to sentences in one of four genres: newswire (NW), web text (WT), broadcast news (BN), and broadcast conversations (BC). Answers are judged "*correct*" only if they either correctly identify a null answer (i.e., there is no answer to be returned) or correctly extract an answer that is present in the sentence. Answers are not penalized for including extra text *excluding* text from another answer or text from another top-level predicate.

2.1. UMD Parser

We have found that training a parser that matches the genre was of critical importance. For this reason, in University of Maryland (UMD) parser, we have re-implemented and enhanced the Berkeley parser [4] in several ways. While the enhancements to the parser are important, it is even more important to train grammars matched to the conditions of use. Therefore, our genre-specific English grammars were trained by subsampling from available treebanks (WSJ, BN, Brown, Fisher, and Switchboard) after some genre-specific text transformations. For informal speech, we replaced symbolic expressions with verbal forms and removed punctuation, edits, and case. We also utilized a large amount (210K to 300K sentences) of genre-matched self-labeled training parses in training these grammars. We also trained grammars with a subset of function tags annotated in the treebank that indicate case role information (e.g., SBJ, OBJ, LOC, TMP, MNR) in order to produce function tags. Using these parsers, we obtained F_1 measures of 91.7% on WSJ section 23 training on WSJ (not including dev and test), 90.45% on 10% of the labeled data drawn from English BN, and 87.84% on English Fisher on 10% of the labeled data drawn from Fisher.

2.2. Individual Systems for Answering 5-W Questions

We developed three systems to answer 5-W questions: one at ICSI/SRI, one at NYU, and one at Columbia University. We will refer to these systems as System A, System B, and System C, respectively.

System A works in two steps: The first step is a cascade of several operations to determine one top-level predicate using UMD parser, including detecting and marking quotations, removing conditional clauses, processing conjunction of sentences and conjunction of verb phrases so that only one top-level predicate remains, and detecting passive sentences. The second step starts with an analysis of the sentence structure since the positions of constituents in the sentence depend on the sentence structure. Once the sentence is categorized according to its structure, a set of linguistically-motivated handcrafted rules is applied to extract answers. These rules make use of the bracketed syntactic parse trees with function tags produced by the UMD parser.

System B uses a Grammatical and Logical Argument Framework (GLARF)-based program [5] to recognize logical relations between words, such as subject, object, and indirect object. The GLARF system regularizes these relations across passive, relative clauses, coordinate structures, and so on. The 5Ws are read off of these relations, mapping the logical subject to *who*, the verb plus one core argument to *what*, temporal adverbials to *when*, and so on. 5-W output was calculated for GLARF output of both the Charniak parser [6] and the UMD parser, using heuristics to choose between the two (the one with more non-null answers, etc.). If neither of these systems produced output that included a verb in the WHAT slot, a secondary string-based set of heuristics was used to fill the 5W slots. Typically, the backup system fired only when both parsers failed or produced nonsensical output.

System C was developed specifically for handling the noisy sentences resulting from ASR by a series of fall-back systems. We developed a set of information extraction patterns, trained over speech data, to identify the top level predicate using the dependency tree and function tags produced by the UMD parser tuned for speech. If a unique predicate was found, then the function tags, along with time and location named entities, were used to extract the 5Ws from the sentence. If no unique predicate was found, then a disfluency removal algorithm was applied and the same IE patterns were then applied to the dependency tree produced by the Stanford Parser to identify the top-level predicate and its arguments. If this method failed, then System C falls back to the use of NYU's Jet system to perform chunking and a different set of information extraction patterns.

3. Combining Multiple Answers

When only automatically generated syntactic parse trees are available, the quality of the available information substantially degrades. The automatic transcriptions of audio input present many challenges as they do not contain punctuation, include disfluencies, and often have segmentation issues. Under such real-world conditions, a combination of several modules each of which presents a different view provides a mechanism to overcome the shortcomings of individual modules.

3.1. Feature Extraction

It was found that the performance of a 5-W question answering system correlates with some higher-level features, which are summarized in Table 1. These features can be categorized as follows:

- *System Level Features*: The features numbered (1) through (6) encode system-specific answer construction mechanism. For instance, a system might be failing in answering *who* in passive sentences that can be detected by a "non-null answer to *who*" (feature (2)).
- *System Agreement Features*: Features of type (7) compare answers from different systems. The rationale for these features is that if any two systems have overlapping answers, then it is more likely that their answers are correct. In the meantime, if two systems agree on their predicates, then some of their answers should agree or overlap.
- *Syntactic Features*: The features numbered (8) capture whether one should be expecting a locative, temporal, or causative argument to be non-null.
- *Sentence Level Features*: The features numbered (9) and (10) are found by analyzing the given sentence itself.

System level features are extracted from each of the three systems separately as these depend on the system. These features are then concatenated with other features and used to train classifiers.

3.2. Training Discriminative Classifiers

The answers are selected from one system only depending on *some prediction score that indicates how likely it is that a selected system will return as many correct answers as possible*. Since it is possible more than one system to be the highest scoring, this is a *multiclass multilabel classification* problem.

Since the arguments are dependent on the predicate chosen, we do not mix and match fields. Even when two systems select the same top-level predicate, their answers might be different, because, in some cases, constituents may answer more than one question. For instance, for the sentence "*She was depressed after the car accident*", the constituent "*after the car accident*" is allowed to answer either one of the *when* and

Table 1: The rationale of features extracted for combining multiple 5-W systems. Classification-based approach takes system level features, sentence level features, system agrement features, and syntactic features to combine multiple systems at the granularity of sentences or at the granularity of individual fields.

FEATURE TYPE	RATIONALE
(1) Number of non-null answers	A system returning fewer non-null answers is likely to be answering all 5-W questions.
(2) Positions of the answers	The position of an answer should not be too far from the positions of other answers.
(3) Number of words in each answer	The length of an answer is typically dependent on the field; for instance, the length of the answer to *what* is typically greater than one as the correct answer to *what* should include the main predicate *and* the object (if any).
(4) The answer to *who* is null	There are cases when the answer to *who* should be null (e.g., imperative sentences) and other cases when it should not be null (e.g., simple declarative sentences).
(5) WordNet analysis of the answer to *who*	The answer to *who* should include a noun.
(6) WordNet analysis of the answer to *what*	The answer to *what* should include a verb, and typically a noun; it typically starts with a verb and the verb comes before a noun.
(7) Agreement among systems	The answers to *who, what, when, where,* and *why* as well as the predicates would agree when all the systems consider the same predicate. If predicates do not agree, answers should not be expected to agree.
(8) Number of arguments in the sentence	The number of times "LOC", "TMP", "PRP" arguments occur in the sentence evidences if one should expect to have non-null answers.
(9) Voice of the sentence	The answers to *who* and *what* depend on whether the so-found top-level predicate is in its active or passive form.
(10) Length of the sentence	Answers are typically more reliable for short sentences.
(11) Quotations	The quotations should appear as they are.

why questions. Furthermore, as our experimental results show, the systems we developed make complementary mistakes and therefore the ideal combination strategy that selects the best system for each sentence was able to give an accuracy of as much as 96% overall.

3.2.1. 5-W Corpus

Since each sentence may have multiple correct sets of 5-Ws, it is not straightforward to produce a gold-standard corpus for automatic evaluation. One would have to specify answers for each possible top-level predicate, as well as which parts of the sentence are optional and which are not allowed. This also makes creating training data for system development problematic and expensive.

In this work, we selected the first five (reasonable) sentences from 20 English documents. The answers for these 100 sentences were presented side by side in a Java-based interface and graded by human judges. The interface showed the sentence, the system responses, and options for selecting "*correct*","*incorrect*", and "*partial*" options for each answer. For BN and BC sentences, the graders were presented the manual and automatic transcriptions, as the evaluation should be performed against the manual transcription but the answers should be extracted from the automatic transcriptions. Each sentence was graded by two human judges, and disagreements were resolved by further discussions. Since the annotated data we collected was not large, we ran a 20-fold cross-validation and take average in our experiments.

3.2.2. System Level Combination

In this combination strategy, we selected training binary SVM classifiers[1] to solve this multiclass multilabel classification problem. Three binary classifiers, called SVM_A, SVM_B and SVM_C, predict how likely that system $s \in \{A, B, C\}$ would return as many correct answers as possible for a given sentence.

Let x denote the concatenation of the features extracted from the answers of the three systems, from the sentence itself, and from its syntactic analysis as described in Section 3.2. Let y_s denote the labels that are used to train SVM_s. The labels, y_s, are such that

$$y_s = \begin{cases} 1 & \text{, if System } s \text{ is a highest scoring system,} \\ 0 & \text{, otherwise.} \end{cases} \quad (1)$$

In the test stage, the scores from the three SVMs are compared, the system with the highest prediction score is selected, and its answers are returned.

3.2.3. Answer Level Combination

In this combination strategy, the responses of those systems that received a "*correct*" grade for a target question are labeled with a "1", and those of lower-scoring systems are labeled with a "0". The feature vectors are used to train five binary SVMs, one for each of the 5-W questions. Each of these SVMs predicts whether a given answer (rather than a given system) is correct or not. Let x denote feature vectors as before and let z_s^q denote the binary labels attached to the answers of System $s \in \{A, B, C\}$ to the question $q \in \{who, what, when, where, why\}$. The feature vectors are labeled such that

$$z_s^q = \begin{cases} 1 & \text{, if System } s\text{'s answer to } q \text{ is correct,} \\ 0 & \text{, otherwise.} \end{cases} \quad (2)$$

The prediction scores of binary SVMs are then used to form new feature vectors to train three binary SVMs to predict the best system for the given sentence. These feature vectors, $\hat{p}_s$, are composed of the sums of the prediction scores of System s to the 5-W questions. The corresponding labels are such that if System s is a highest-scoring system, its label is 1, and 0 otherwise. The final decision is made by comparing all three prediction scores and selecting the highest one.

4. Experiments and Results

We ran experiments to evaluate the two combination strategies and compare their performance against individual systems as well as combination strategies that would be taken in case no annotated data was available.

4.1. Performance Evaluation Per Genre

Table 2 compares the error rates of the different systems and combination strategies on different genres. "Random" stands for a strategy in which the answers of a randomly selected system are returned. "Oracle" stands for the ideal strategy that

[1] http://svmlight.joachims.org/

Table 2: Error rates per genre. The answer level combination reduces the error rate of the majority-voting based combination by 17% to 35%.

	NW	WT	BN	BC	Pooled
Sys A	14.8	17.0	12.0	12.8	14.2
Sys B	12.6	17.1	15.9	15.9	15.4
Sys C	22.0	22.8	9.2	11.8	16.2
Random	16.2	19.5	12.1	14.7	15.6
Majority	16.4	16.3	11.1	15.2	14.7
Oracle	3.9	7.2	2.7	2.2	4.0
System Level	13.0	**13.3**	9.1	12.6	12.0
Answer Level	**10.6**	13.6	**8.0**	**11.4**	**10.9**
δ_{Random}	36.1	30.6	33.9	22.7	30.5
$\delta_{Majority}$	35.4	16.6	27.9	25.0	25.9
δ_{Best}	15.87	20.0	13.0	3.4	23.2

has the knowledge of which system would perform best for a given sentence, and hence denotes the best that can be expected from any combination scheme. δ_{Random} stands for the relative improvement of the answer level combination strategy over "Random", $\delta_{majority}$ stands for that over "Majority", and δ_{best} stands for that over the best individual system. The column "Pooled"' shows the error rate when answers from all genres and all question types are considered.

If we had no annotated data and hence no knowledge about how these individual systems perform, we could take two approaches. As a first approach, we could just select a system randomly for each given sentence and return its answers. This totally "ignorant" system combination would correspond to the "Random" strategy. As an alternative, we could take a more intelligent strategy and test if any systems agree on any answer. If there is any agreement, for each sentence, we could score any agreement with a "+1" and then return the answers of the highest-scoring system. This combination would correspond to the "Majority" strategy in Table 2.

As seen in Table 2, the answer level combination strategy outperformed the "Random" strategy by 22.7% to 36.1% relative and the "Majority" strategy by 16.6% to 35.4% relative. If some annotated data were made available, then we could evaluate the performance of individual systems and select the best one for each genre. As seen in Table 2, the best system in NW is System B, in WT it is System A, and in BN and BC it is System C. The answer level combination strategy was able to improve all these systems and, more specifically, the best individual system by 3.4% to 20% relative. The system level combination performed slightly better than answer level combination in WT but the difference is not significant.

4.2. Performance Evaluation Per Question Type

As seen in Table 3, the performance on the answers to *what* is substantially lower than those answers to other questions. One of the most important reasons is that the answers to *what* tend to contain extra text, which quite often contains arguments of a predicate other than the top-level predicate. It was also quite common that the answer to *what* contained text that would correctly answer other questions. For instance, if a passive sentence detector fails, then the answer to *who* is "incorrect" and the answer to *what* is "partial" at best, and similarly, if the answer to *when* is contained in the answer to what, then the answers to both *when* and *what* are "incorrect".

Table 3 shows that the answer level combination strategy was able to significantly improve all "Random", "Majority", and "Best system" strategies. These improvements were most obvious in the answers to *what*, suggesting us to conclude that the most dramatic improvement would in the worst-performing field.

Table 3: Error rates per question type. The answer level combination reduces the error rate of the majority-voting based combination by 16% to 40%.

	who	what	when	where	why
Sys A	12.6	23.8	11.8	16.7	5.3
Sys B	15.8	30.9	13.7	10.8	4.9
Sys C	22.3	26.4	21.0	13.1	9.4
Random	14.1	28.2	16.1	13.8	5.4
Majority	13.7	24.1	13.1	14.0	6.0
Oracle	4.6	6.4	9.5	9.5	2.3
System Level	10.0	21.3	11.5	12.6	4.6
Answer Level	**8.2**	**18.6**	**11.0**	**11.0**	**4.4**
δ_{Random}	41.8	34.0	31.8	19.7	19.1
$\delta_{Majority}$	40.1	22.8	16.0	21.4	26.7
δ_{Best}	33.4	21.8	6.5	33.9	17.1

A major conclusion that we make from Tables 2 and 3 is that the combination strategy that tackles the problem in the granularity of individual fields is more successful than the strategy that works in the granularity of sentences. A similar observation was made in combining systems for semantic role labeling as well [3].

5. Conclusions

We describe two combination strategies, one of which operates at the granularity of sentences and the other at the granularity of answers. The specific task we addressed was the 5-W in which the goal is to answer (*who, what, when, where,* and *why*) questions for each given sentence. We trained SVM classifiers with a set of novel features. We evaluated the proposed strategies using a set of text and audio sentences. Our experimental results indicated that the proposed features and combination strategies were successful at utilizing the strengths of each component system. The combination strategies described in this paper can be used to make effective use of multiple answers originating from independent systems in information extraction, semantic role labeling and etc.

6. Acknowledgements

The authors thank Sara Stolbach for developing a graphical user interface for data annotation and Bob Coyne for participating in data annotation. This work was supported by DARPA HR0011-06-C-0023. Any opinions, findings and/or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

7. References

[1] E. W. D. Whittaker, J. Mrozinski, and S. Furui, "Factoid question answering with web, mobile and speech interfaces," in *NAACL/HLT*, 2006.

[2] L. Lamel, S. Rosset, C. Ayache, D. Mostefa, J. Turmo, and P. Comas, "Question answering on speech transcriptions: the QAST evaluation in CLEFF," in *LREC*, 2008.

[3] M. Surdeanu, L. Marquez, X. Carreras, and P. R. Comas, "Combination strategies for semantic role labeling," *Journal of Artificial Intelligence Research*, vol. 29, pp. 105–151, 2007.

[4] S. Petrov and D. Klein, "Improved inference for unlexicalized parsing," in *NAACL/HLT*, 2007.

[5] A. Meyers, M. Kosaka, N. Xue, H. Ji, A. Sun, S.Liao, and W. Xu, "Automatic Recognition of Logical Relations for English, Chinese and Japanese," in *SEW-2009 at NAACL-HLT-2009*, 2009.

[6] E. Charniak, "Immediate-head parsing for language models," in *Meeting of the ACL*, 2001.