Identifying Opinion Holders for Question Answering in Opinion Texts

Soo-Min Kim and Eduard Hovy

Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292-6695
{skim, hovy}@isi.edu

Abstract

Question answering in opinion texts has so far mostly concentrated on the identification of opinions and on analyzing the sentiment expressed in opinions. In this paper, we address another important part of Question Answering (QA) in opinion texts: finding opinion holders. Holder identification is a central part of full opinion identification and can be used independently to answer several opinion questions such as "Is China supporting Bush's war on Iraq?" and "Do Iraqi people want U.S. troops in their soil?". Our system automatically learns the syntactic features signaling opinion holders using a Maximum Entropy ranking algorithm trained on human annotated data. Using syntactic parsing features, our system achieved 64% accuracy on identifying the holder of opinions in the MPQA dataset.

Introduction

Recently, the problem of detecting opinions in text has been studied by many researchers. This work promises to have important impact in the question answering community. For general QA, beyond question-types such as "What does X think of Y", opinion detection is important to determine whether the answer to a question is a fact or just someone's opinion.

In opinion domains, several types of questions must be answered, such as: "What is people's opinion about Y", "What does X like?", and "Who strongly believes in Y". Examples of such restricted domains include customer product feedback, movie reviews, editorials, as well as blogs and newsgroups focusing on topics like public opinions on social issues and political events.

Various approaches have been adopted to address the first two types of questions. Pang et al. (2002) and Turney (2002) classified sentiment polarity of reviews at the document level. Wiebe et al. (1999) classified sentence level subjectivity using syntactic classes such as adjectives, pronouns and modal verbs as features. Riloff and Wiebe (2003) extracted subjective expressions from sentences using a bootstrapping pattern learning process. Kim and Hovy (2004) automatically generated subjective unigrams using WordNet and then used them as clues to recognize opinion-bearing sentences. Yu and Hatzivassiloglou identified the polarity of opinion sentences using semantically oriented words. The Text Retrieval Conference (TREC) also held a task of finding relevant opinion sentences to a given topic as a part of Novelty track (2002-2004).

Answering questions such as "Who strongly believes in Y" requires a system to recognize the *holder* of opinion Y. Despite the successes in identifying opinion expressions and subjective words/phrases, researchers have been less successful at identifying the factors closely related to subjectivity and polarity, such as opinion holder, topic of opinion, and inter-topic/inter-opinion relationships. By detecting opinion holders, we can answer questions such as "Is China supporting Bush's war on Iraq?", "Which European countries are against the war on Iraq?, "How do American people think about tax cut" or "Do Iraqi people want U.S. troops in their soil?". Recognizing the opinion holder is important in order to know how people think about social or public issues in making policies and surveying public opinions. Stock market predictors are interested in what people feel about certain products and companies, and manufacturers, advertising agencies, and the film industry cares about public ratings of their products. Especially on news group message boards or on governmental web sites, many people express their opinion about controversial issues so that they can participate more actively in making rules. It is also critical to know how different countries think about a political event to deal with international relations. By grouping opinion holders like countries or president of each country, we can potentially have better understanding of international relationships.

In this paper, we propose an automatic method for identifying opinion holders. We define the opinion holder as an entity (person, country, organization, or special group of people) who expresses explicitly or implicitly the opinion contained in a sentence. We first identify all possible opinion holder entities in a sentence and apply the Maximum Entropy ranking algorithm to select the most probable one.

Identifying opinion holders is difficult especially when the opinion sentence contains more than one likely holder entity. In the example sentence "*Russia's defense minister said Sunday that his country disagrees with the U.S. view of Iraq, Iran and North Korea as an 'axis of evil'*", candidate holders for the reported opinion "*disagrees with the U.S. view*" are "*Russia*", "*Russia's defense minister*", "*U.S.*", "*Iraq*", "*Iran*", "*North Korea*". Another difficult problem occurs when there is more than one opinion in a sentence. In that case, we have to find the right holder for each opinion. For example, in "*In relation to Bush's axis of evil remarks, the German Foreign Minister also said, Allies are not satellites, and the French Foreign Minister caustically criticized that the United States' unilateral, simplistic worldview poses a new threat to the world*", "*the German Foreign Minister*" should be the holder for the opinion "*Allies are not satellites*" and "*the French Foreign Minister*" should be the holder for "*caustically criticized*".

This paper is organized as follows. In the next section, we introduce the data we used for our study. We then describe our machine learning approach with an explanation of the feature selection. We report system experiments and results and conclude in the last section.

Data

As training data, we used the MPQA[1] corpus (Wiebe et al., 2003) that contains news articles manually annotated by 5 trained annotators using an annotation scheme for opinions. This corpus consists of 10657 sentences from 535 documents, annotated for four different aspects: *agent, expressive-subjectivity, on,* and *inside*. *Expressive-subjectivity* marks words and phrases that indirectly express a *private state* that is defined as a term for opinions, evaluations, emotions, and speculations. *On* annotation marks speech events and direct expressions of private states. Both of them have strength attributes that indicate the strength of private state. For our task, we only selected expressions with high strength (*high* or *extreme*) since expression with low strength will likely have lower inter-annotator agreement (Wilson and Wiebe, 2003)[2]. As for the holder, we use the agent of the selected private states or speech events. Table 1 shows an example of the annotation. In this example, we consider the expression "the U.S. government 'is the source of evil' in the world" with an expressive-subjectivity tag as an opinion of the holder "Iraqi Vice President Taha Yassin Ramadan" since it is annotated with the strength "extreme."

[1] http://www.cs.pitt.edu/~wiebe/pubs/ardasummer02/

[2] Expressive-subjectivity with strength of high or extereme is reported to have 0.88 inter-annotator agreement, whereas expressive-subjectivity with strength of medium, high, or extreme has only 0.80.

Approach

Since more than one opinion may be expressed in a sentence, it is not enough simply to pick one holder per sentence. We have to find an opinion holder for each opinion expression. For example, in a sentence "A think B's criticism of T is wrong", B is the holder of "the criticism of T", whereas A is the person who has an opinion that B's criticism is wrong. Therefore, we define our task as finding an opinion holder, given an opinion expression in a sentence. Our earlier work (Kim and Hovy, 2004) focused on identifying opinion expressions within text. We employ that system in tandem with the one described here.

Sentence	Iraqi Vice President Taha Yassin Ramadan, responding to Bush's 'axis of evil' remark, said the U.S. government 'is the source of evil' in the world.
Expressive subjectivity	the U.S. government 'is the source of evil' in the world
Strength	Extreme
Source	Iraqi Vice President Taha Yassin Ramadan

Table 1. Annotation example.

Maximum Entropy Ranking

To learn opinion holders automatically, we use Maximum Entropy. Maximum Entropy models implement the intuition that the best model is the one that is consistent with the set of constraints imposed by the evidence but otherwise is as uniform as possible (Berger et al. 1996). There are two ways to model the problem with ME: classification and ranking. Classification allocates each holder candidate to one of a set of predefined classes while ranking selects a single candidate as answer. This means that classification modeling[3] can select many candidates as answers as long as they are marked as true, and does not select any candidate if every one is marked as false. In contrast, ranking always selects the most probable candidate as an answer, which suits our task better. Our earlier experiments showed poor performance with classification modeling, an experience also reported for the question answering task (Ravichandran et al. 2003).

We modeled the problem to choose the most probable candidate that maximizes a given conditional probability distribution, given a set of holder candidates $\{h_1 h_2 \ldots h_N\}$ and opinion expression e. The conditional

[3] In our task, there are two classes, holder or not, in classification modeling.

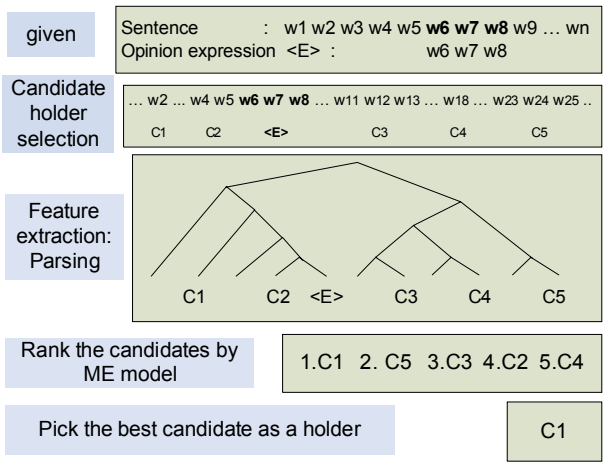

Figure 1. Overall system architecture.

probability $P(h|\{h_1 h_2 \ldots h_N\}, e)$ can be calculated based on K feature functions $f_k(h, \{h_1 h_2 \ldots h_N\}, e)$. We write a decision rule for the ranking as follows:

$$\hat{h} = \arg_h [P(h \mid \{h_1 h_2 \ldots h_N\}, e)]$$
$$= \arg_h [\sum_{k=1}^{K} \lambda_k f_k(h, \{h_1 h_2 \ldots h_N\}, e)]$$

Each λ_k is a model parameter indicating the weight for its feature function.

Overall System Architecture

Figure 1 describes our holder identification system. First, the system generates all possible holder candidates, given a sentence and an opinion expression <E>. After parsing the sentence, it extracts features such as syntactic path information between each candidate <H> and the expression <E> and a distance between <H> and <E>. Then it ranks holder candidates according to the score obtained by the ME ranking model. Finally the system picks the best candidate with a highest score. The following sections describe how to select holder candidates and how to select features for the training model.

Holder Candidate

Intuitively, one would expect most opinion holders to be named entities (PERSON or ORGANIZATION). However, also common noun phrases can be opinion holders, such as "the U.S leader", "Iranian officials", and "the Arab and Islamic world". Sometimes, pronouns like *he, she,* and *they* that indicate a PERSON or *it* that indicates an ORGANIZATION or country can be an opinion holder. In our study, however, we do not consider pronoun holders for several reasons. First, even if we pick the correct pronoun holder, say "he", this does not really provide the requisite information until we determine what "he" refers to. If a more specific entity than the pronoun appears in the sentence, we should pick it by not letting ME consider pronoun holders. Second, for most cases with a pronoun as the holder, the referent named entity appears in some previous sentence. Solving the co-reference resolution problem is beyond the boundary of our work. As a result, we considered only named entities and noun phrases as holder candidates.

Table 2 shows an example of all noun phrase and named entity holder candidates selected for the sentence in Table 1. We use BBN's named entity tagger IdentiFinder to collect named entities and Charniak's parser to extract noun phrases.

Candidate	Type
Taha Yassin Ramadan	PERSON
Bush	PERSON
U.S.	LOCATION
Bush's	NP
Bush's 'axis of evil' remark	NP
the world	NP
evil'	NP
evil' in the world	NP
The source	NP
The source of evil' in the world	NP
The U.S. government	NP
Iraqi Vice President Taha Yassin Ramadan	NP

Table 2: Candidates selected from the sentence in Table 1.

Feature Selection

We describe three types of features in this section: full parsing features, partial parsing features, and others. Our hypothesis is that there exists a structural relation between a holder <H> and an expression <E> that can help to identify opinion holders. This relation may be represented by lexical level patterns between <H> and <E>, but anchoring on specific words might run into the data sparseness problem. For example, if we see the lexical pattern "<H> recently criticized <E>" in the training data, it is impossible to match the expression "<H> yesterday condemned <E>". To determine how much generalization is needed beyond the lexical level, and hence to prove our hypothesis, we selected structural features from a deep parse, a partial parse, and the surface level, and used ME to compare performances. For deep parsing we used the Charniak parser and for partial parsing the CASS parser[4].

[4] http://www.sfs.nphil.uni-tuebingen.de/Staff-Old/abney/#cass

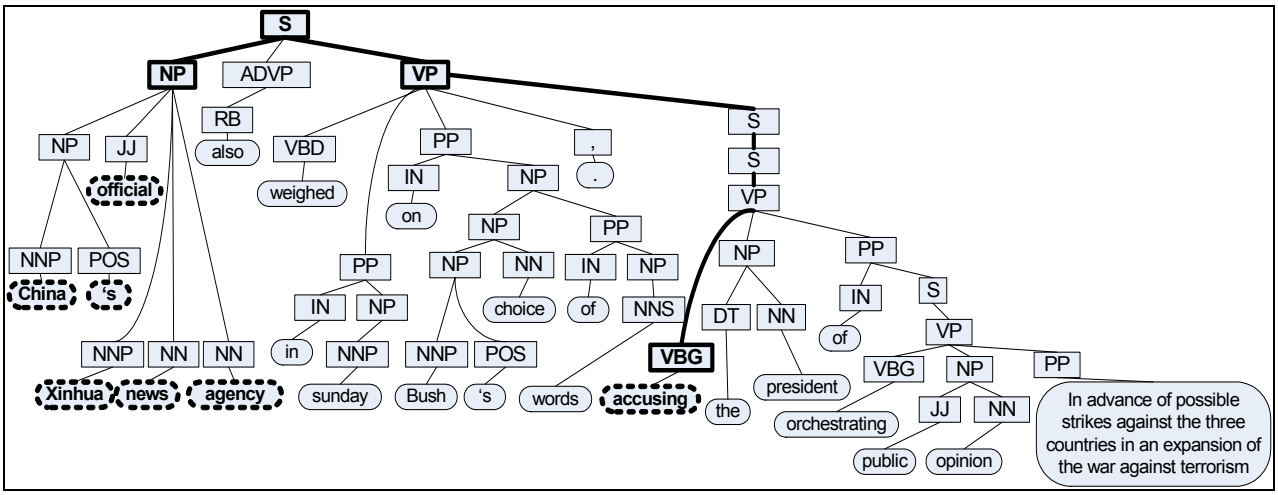

Figure 2. A parsing example.

Parsing Features

After parsing the sentence, we search for the *lowest common parent node* of the words in <H> and <E> respectively (<H> and <E> are mostly expressed with multiple words). A lowest common parent node is a non-terminal node in a parse tree that covers all the words in <H> and <E>. Figure 2 shows a parsed example of a sentence with the holder "China's official Xinhua news agency" and the opinion expression "accusing". In this example, the lowest common parent of words in <H> is the bold NP and the lowest common parent of <E> is the bold VBG. We name these nodes *Hhead* and *Ehead* respectively. After finding these nodes, we label them by subscript (e.g., NP_H and VBG_E) to indicate they cover <H> and <E>.

In order to see how Hhead and Ehead are related to each other in the parse tree, we define another node, *HEhead,* that covers both Hhead and Ehead. In the example, HEhead is S at the top of the parse tree since it covers both NP_H and VBG_E. We also label S by subscript as S_{HE}.

To express tree structure for ME training, we extract path information between <H> and <E>. In the example, the complete path from Hhead to Ehead is "<H> NP S VP S S VP VBG <E>". However, representing each complete path as a single feature produces so many different paths with low frequencies that the ME system would learn poorly. Therefore, we split the path into three parts, as in Table 3. With this splitting, the system can work when any of HEpath, Hpath or Epath appeared in the training data, even if the entire path from <H> to <E> is unseen.

Table 4 summarizes these concepts with two holder candidate examples in the parse tree of Figure 2. Among the children nodes of HEhead, we ignore any other nodes that do not relate to <H> or <E>. But these paths are sometimes so long that they may still encounter the data sparseness problem. This motivated us to examine path generalization. We add another feature that only considers the top two levels below a child node of HEhead on the path toward Hhead. With this feature, we can consider the paths "<H> NP_H PP_H NP_H " and "<H> NP_H NP_H PP_H VP_H NP_H PP_H NP_H" as the same because they share "PP_H NP_H" at the top.

Path	From	To
HEpath	HEhead	Two children nodes that are also parents nods of Hhead and Ehead
Hpath	Hhead	Hhead's ancestor node that is a child of HEhead
Epath	Ehead	Ehead's ancestor node that is a child of HEhead

Table 3. Path definition.

	Candidate 1	Candidate 2
	China's official Xinuhua news agency	Bush
Hhead	NP_H	NNP_H
Ehead	VBG_E	VBG_E
HEhead	S_{HE}	VP_{HE}
Hpath	NP_H	NNP_H NP_H NP_H NP_H PP_H
Epath	VBG_E VP_E S_E S_E VP_E	VBG_E VP_E S_E S_E
HEpath	S_{HE} NP_H VP_E	VP_{HE} PP_H S_E

Table 4. Heads and paths example.

Partial Parsing Features

Full parsing provides rich information about a sentence structure but it often produces too large and deep parse trees. It raises a question about the necessity of full

parsing. Therefore, we applied a partial parser and extracted features from its result to compare its performance with full parsing. Figure 3 shows a chunk example of the partial parser.

(c **China's official Xinhua news agency** also weighed in Sunday on Bush's choice of words,)
(vgp **accusing** the president of orchestrating public opinion in advance of possible strikes against the three countries in an expansion of the war against terrorism.)

Figure 3. Partial parsing example.

As features we use the tags of the chunk containing <H> and of the chunk containing <E>. In the following example, the chunk tag for the holder **"**China's official Xinhua news agency**"** is *c* and the tag for the opinion expression "accusing" is *vgp*. We also consider whether <H> and <E> belong to the same chunk.

Other Features

We also include two non-structural features. The first is the type of the candidate, with values NP, PERSON, ORGANIZATION, and LOCATION. This feature enables ME to determine the most probable one among them automatically. The second feature is the distance between <H> and <E>, counted in parse tree words. This is motivated by the intuition that holder candidates tend to lie closer to their opinion expression. All features are listed in Table 5.

Experiments and Results

Answer Selection

Choosing an answer (or answers) from multiple holder candidates is an important issue for training and evaluating. Even though each opinion expression has been annotated with a holder in the training and test sentences, we cannot consider only candidates that match *exactly* with the marked holder. In many cases, just part of the holder is enough for a valid answer. For example, when a holder is "Michel Sidibe, Director of the Country and Regional Support Department of UNAIDS", just "Michel Sidibe" is also enough to be an answer. Or, in the case "Chief Minister Dr. Farooq Abdullah", the title part "Chief Minister" and the rest "Dr. Farooq Abdullah" are both acceptable answers along with the whole phrase.

On the other hand, just allowing any holder candidate that partially matches with the holder does not work. For example, given a holder "the head of the medical staff of Amnesty International, Jim West", noun phrases like "the head" or "medical staff" are not representative enough for an answer. If a more proper holder candidate like "Jim West" exists, we should pick it. In order to generate the gold standard, we assign each candidate a priority value first that represents how likely it is to be an answer. Table 6 shows the priority value assignment algorithm.

Features	Description
f1	Type of <H>
f2	HEpath
f3	Hpath
f4	Epath
f5	Distance between <H> and <E>
f6	Top two levels of Hpath
f7	Chunk tag of <E>
f8	Chunk tag of <H>
f9	Whether <E> and <H> are in the same chunk

Table 5. Features for ME training.

Priority	Condition
1	|Overlaped string| > threshold1 and |Irrelevant words| < threshold2
2	|Overlapped string| > threshold1
3	|Overlapped string| > 0

Table 6. Priority value assignment algorithm for answer selection.

We use 0.5 for threshold1, which means we allow a candidate as an answer in case half of the words in a holder appear in the candidate as well. With this threshold, given a holder such as "President Bush", we allow both "President" and "Bush" as eligible answers. However, if a candidate contains many irrelevant words, it is less likely to be an answer than any candidate that contains only relevant words. We use threshold2 for this purpose and assign it 4, since the average number words in human annotated holders is 3.71.

After assigning these values to candidates, we pick candidates in priority order as answers. Using this algorithm, only 155 candidates among the total 1078 candidates in the test data (14%) are picked. This means that on average 11 candidates are picked by the system for each case but only 1.58 of them are marked correct on average. We call this selection method a *strict selection*. However, when we accept candidates with any priority (1,2, or 3), 36% of candidates are marked correct. We call this a *lenient selection*. We report experimental results of both selections below.

Evaluation Result

From the MPQA data described above, we collected 961 pairs of (<E>,<H>) and divided them into a training set of 863 and a test set of 98. We evaluated system performance by accuracy: the number of times a correct holder was found divided by the total number of cases.

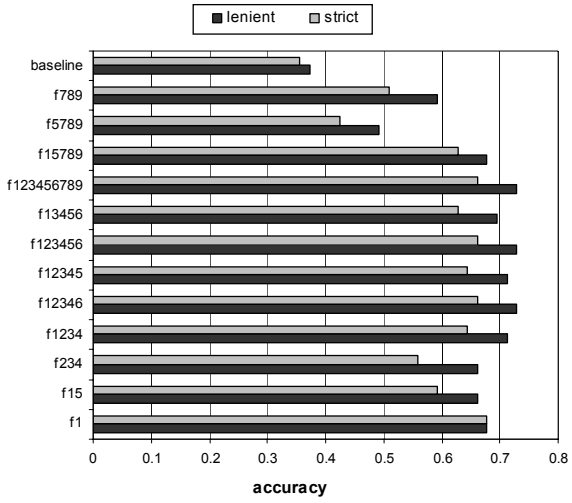

Figure 4. Named Entity candidates.

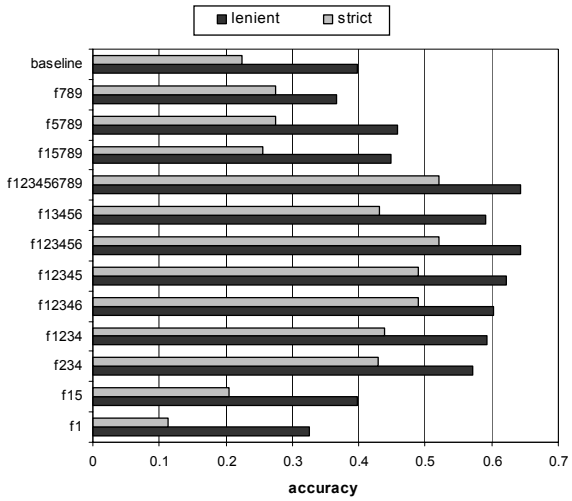

Figure 5. Noun Phrase candidates.

We provided candidate holders for ME choice using two methods: just named entities or all NPs[5]. Since named entities constitute only 59% of the correct answers in the test data, for the named entity evaluation we manually extracted and used only those sentences. Thus for evaluation purposes, we picked those 59% for the named entity candidate selection method but used the whole test data set for the NP candidate selection.

Figures 4 and 5 report the accuracy of each method. For notational convenience, we represent each feature by its index on the axis. The average number of candidates selected is 2.95 for NE selection and 11 for NP selection.

To evaluate the effectiveness of our system, we set the baseline as the system choosing the closest candidate to the expression as a holder without ME decision. This baseline system performed only 0.39 accuracy (lenient measure) and 0.22 (strict measure) in Figure 5. Among the 9

[5] Noun phrases include named entities.

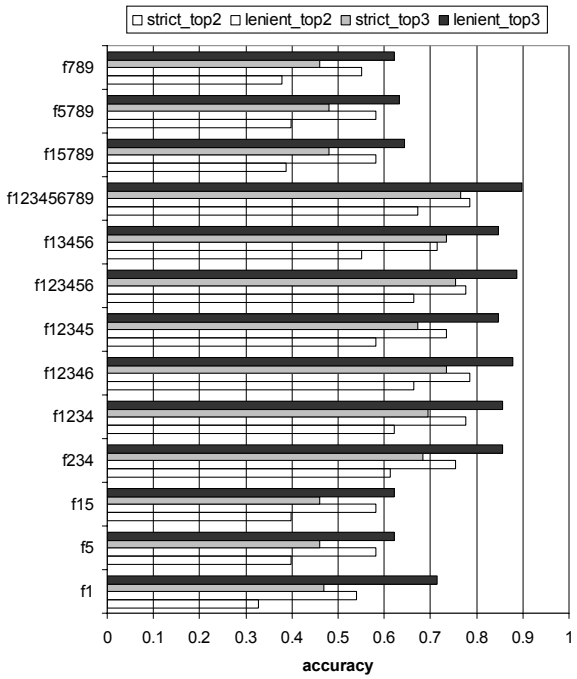

Figure 6. Accuracy of top 2 and top 3 answers.

features, we group f2, f3 and f4 together since they reflect deep parsing features and group f7, f8 and f9 together as chunk features for experimental convenience.

As can be seen from the results in Figures 4 and 5, features other than distance or candidate type helped in performance. Especially the structural features of parsing, f2, f3, and f4, improved the performance significantly. We interpret these results as demonstrating that there exists a clear syntactic relationship between a holder and an opinion expression. Another observation we found from this study is that partial parsing (features f789) does not adequately capture the holder and expression relationships. Figure 5 shows that deep parsing features (f234) perform at 0.57 but partial parsing (f789) perform at only 0.36 in lenient measure. Similarly, deep parsing features combined with f1 and f5 perform at 0.62 but partial parsing features combined with the same features perform at 0.44. Feature f6 (taking only the top two levels of Hpath) performs worse than the full path feature (compare f13456 to f12345).

To evaluate our system more practically, we counted how many test data found correct answers within top 2 and top 3 answers system provided instead of just top first answer. Figure 6 shows accuracy of noun phrase candidate selection in strict and lenient measure on the whole test data. Again the structural features helped more than any other feature, reaching up to 90% of accuracy in lenient measure.

Table 7 shows the system output of the example in Figure 2. The system successfully finds the right holder as

the highest ranked answer, and ranks the candidate "Bush" at the bottom.

Rank	Candidates	Score
1	China's official Xinhua news agency	0.053999
2	Xinhua	0.053972
3	China's	0.051066
4	China	0.051053
5	the president	0.048306
6	possible strikes	0.048274
7	public opinion	0.048180
8	the three countries	0.048172
9	Advance	0.048127
10	possible strikes against the three countries	0.048110
11	an expansion	0.047937
12	the war against terrorism	0.047812
13	an expansion of the war against terrorism	0.047751
14	the war	0.047595
15	Terrorism	0.047542
16	Bush's choice of words	0.044291
17	Sunday	0.044136
18	Words	0.043868
19	Bush's choice	0.043697
20	Bush's	0.043073
21	Bush	0.043042

Table 7. System output of the example in Figure 2.

Conclusions

This study describes the automated identification of the holder of a given opinion expression for question answering in opinion text domain. The importance of opinion holder identification was noticed yet it has not been much studied to date, partly because of the lack of annotated data. For our study, we extracted from the MPQA dataset strong opinion expressions on which annotators highly agreed, and for which the opinion holder appeared in the same sentence. We used Maximum Entropy ranking to select the most probable holder among multiple candidates. Adopting parsing features significantly improved system performance. The best feature combination performed at 64% accuracy.

For future work, we plan to experiment with cross-sentence holder identification, in which an opinion holder and an opinion expression may appear in different sentences. We also plan to investigate other types of QA in opinion texts such as automatic topic identification, and the determination of relationships between topics and subtopics in opinion-bearing texts.

References

Abney, S. 1997. *The SCOL Manual: Version 0.1b*. http://www.sfs.nphil.uni-tuebingen.de/Staff-Old/abney /#cass

Berger, A, S. Della Pietra, and V. Della Pietra. 1996. A Maximum Entropy Approach to Natural Language *Computational Linguistics* 22(1).

Charniak, E. 2000. A Maximum-Entropy-Inspired Parser. *Proceedings of NAACL-2000.*

Kim, S. and E.H. Hovy. 2004. Determining the Sentiment of Opinions. *Proceedings of COLING-04*.

Mitchell, T. 1997. *Machine Learning*. McGraw-Hill International Editions: New York, NY, 143–145.

Och, F.J. 2002. Yet Another MaxEnt Toolkit: YASMET http://wasserstoff.informatik.rwth-aachen.de/Colleag ues/och/

Pang, B, L. Lee, and S. Vaithyanathan. 2001. Thumbs up? Sentiment Classification using Machine Learning Techniques. *Proceedings of EMNLP 2002.*

Ravichandran, D., E.H. Hovy, and F.J. Och. 2003. Statistical QA — classifier vs re-ranker: What's the difference? *Proc. of the ACL Workshop on Multilingual Summarization and Question Answering.*

Riloff , E. and J. Wiebe. 2003. Learning Extraction Patterns for Subjective Expressions. *Proceedings of the EMNLP-03*.

Riloff, E., J. Wiebe, and T. Wilson. 2003. Learning Subjective Nouns Using Extraction Pattern Bootstrapping. *Proceedings of CoNLL-03*

Turney, P. 2002. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. *Proceedings of the 40th Annual Meeting of the ACL*, Philadelphia, 417–424.

Wiebe, J., E. Breck, C. Buckley, C. Cardie, P. Davis, B. Fraser, D. Litman, D. Pierce, E. Riloff, T. Wilson, D. Day, D., and M. Maybury. 2003. Recognizing and Organizing Opinions Expressed in the World Press. *Proceedings of the AAAI Spring Symposium on New Directions in Question Answering.*

Wiebe, J, R. Bruce, and T. O'Hara. 1999. Development and use of a gold standard data set for subjectivity classifications. *Proc. of the 37th Annual Meeting of the Association for Computational Linguistics(ACL-99),* 246–253.

Wilson, T. and J. Wiebe. 2003. Annotating Opinions in the World Press. *Proceedings of the 4th SIGdial Workshop on Discourse and Dialogue (SIGdial-03)*.

Wilson, T. and J. Wiebe. 2003. Annotating Opinions in the World Press. *Proceedings of the ACL SIGDIAL-03.*

Yu, H. and V. Hatzivassiloglou. 2003. Towards Answering Opinion Questions: Separating Facts from Opinions and Identifying the Polarity of Opinion Sentences. *Proceedings of the EMNLP conference*.