Recognizing and Organizing Opinions Expressed in the World Press

Janyce Wiebe, Eric Breck, Chris Buckley, Claire Cardie,
Paul Davis, Bruce Fraser, Diane Litman, David Pierce,
Ellen Riloff, Theresa Wilson, David Day, Mark Maybury

Introduction

Tomorrow's question answering systems will need to have the ability to process information about beliefs, opinions, and evaluations—the perspective of an agent. Answers to many simple factual questions—even yes/no questions—are affected by the perspective of the information source. For example, a questioner asking question (1) might be interested to know that, in general, sources in European and North American governments tend to answer "no" to question (1), while sources in African governments tend to answer "yes:"

(1) Was the 2002 election in Zimbabwe fair?

Other questions explicitly ask for information about perspective. For example, consider question (2):

(2) What was the reaction of the U.S. State Department to the 2002 election in Zimbabwe?

In this case, information about the perspective of the U.S. State Department must be identified, both as expressed directly by U.S. State Department spokespeople, and indirectly by other sources.

This paper reports on an exploratory project investigating multiple perspectives in question answering (MPQA). The project was conducted as a summer workshop.[1]

The purposes of this paper are:

- To motivate the need for information about opinions in support of question answering.
- To introduce a framework for annotating, learning, and using information about opinions.
- To demonstrate that information about opinions can be effectively annotated.
- To demonstrate that information about opinions can be effectively learned.
- To formulate a methodology for evaluating the contribution of perspective information to question answering style applications.

The activities of the MPQA project were organized around an end-user task designed to utilize information about perspective—the task of clustering responses to yes/no questions based on perspective. In this task, a questioner may ask a yes/no question (e.g., question (1) above). The system operates as follows: first the question is used as a query to retrieve relevant documents; second, perspective information is identified in the documents; third, passages from the documents are clustered based on their text and perspective features. These clusters are meant to provide an organization of the documents with regard to perspective information to help the questioner understand them.

The remainder of the paper covers the following: The Tasks section discusses the tasks addressed by the MPQA project. The Framework section describes a framework for annotating, learning, and using information about perspective. The Results section reports the results of our preliminary annotation study, machine learning experiments, and clustering experiments. In the annotation study, we found that annotators agreed on about 85**%** of direct expressions of opinion, about 50% of indirect expressions of opinion, and achieved up to 80% kappa agreement on the rhetorical use of perspective. While we will not present the annotation scheme or agreement study in detail, the results demonstrate the feasibility of annotating information about perspective. For machine learning experiments, we trained a very simple classifier for direct expressions of opinion, which achieved 66.4% F-measure, nearly 10% over a baseline system. While we have not yet attempted to learn indirect perspective expressions and other aspects of the annotation scheme, we consider this preliminary result to be an indication of the feasibility of automatic recognition of perspective information. Finally, we evaluated our initial implementation of yes/no clustering with perspective. The results were mixed: for some topics, perspective information helped to cluster "yes" answer passages together quite effectively, while for other topics, the information about perspective did not help. The partial success gives us hope that perspective information will be useful in question answering, but clearly there is a great deal of work to be done.

[1] Funded by the Northeast Regional Research Center (NRRC) of the Advanced Research and Development Activity (ARDA) a U.S. Government entity which sponsors and promotes research of import to the Intelligence Community which includes but is not limited to the CIA, DIA, NSA, NIMA, and NRO.

Tasks

The specific problems addressed by the MPQA project are recognizing and organizing expressions of opinions in the world press and other text. The work builds toward the following tasks to support activities of professional information analysts.

- Given a particular topic, event, or issue, find a range of opinions being expressed about it in the world press.
- Once opinions have been found, cluster them and their sources in useful ways. The *source* of an opinion or perspective is simply the person or group whose opinion or perspective it is. There are various attributes according to which opinions and their sources may be clustered, including:
  - The type of attitude that is expressed. For example, the source might be expressing a positive, negative, or uncertain attitude.
  - The basis for the opinion, such as supporting beliefs, or experiences.
  - The expressive style of the sentences. The style might be sarcastic and vehement, for example, or neutral.
- Once systems are developed to automate the above tasks, they may be applied to many topics and documents, to build perspective profiles of various groups and sources, and observe how attitudes change over time.

To support high-level tasks, such as building perspective profiles over time and recognizing trends and significant changes in opinions, we developed a representation of how opinions are expressed in language, and developed a manual annotation scheme using this representation. The annotation scheme is described in more detail elsewhere. This paper will focus on the overall system architecture and the initial experimental results.

Framework

As part of the MPQA project, we developed a framework for annotating, learning, and using information about perspective. We view this framework as three "architectures" supporting each of these three activities. The annotation architecture supports the annotation of information about opinions in text documents by human annotators. The learning architecture supports the development of automatic perspective recognition components via machine learning. The application architecture supports the yes/no opinion clustering task.

The framework is organized around a database of annotations on documents. In the annotation architecture, human annotators produce annotations of perspective information over the training documents. These training annotations are used in the learning architecture to train system components to automatically identify perspective information in new documents. These components produce annotations of perspective information used by the application architecture to cluster document passages.

A number of general design decisions apply to the annotation database and the MPQA framework as a whole.

- The annotation database implements "standoff", rather than "inline" markup. This means that information about the document is stored separately from the document text. A benefit is that programs only look at the information that they need, without being required to handle a large amount of incidental information.
- Annotation files are considered immutable objects. This means that programs may read annotation files, may write new annotation files, but may never append to existing annotation files.
- The execution model of the framework is "offline" rather than "online". This means that each component of the system may be run separately. A benefit is that modifications to components and updates to the database can be performed without re-building and re-running a large system. (Note that the offline model does not preclude the implementation of a single executable script for running "the system" component by component.)

The remainder of this section briefly describes the design of the annotation, learning, and application architectures of the MPQA framework.

Annotation Architecture

The annotation architecture supports the efforts of human annotators to indicate expressions of opinion in text documents. The primary goal of the architecture is to provide a convenient environment for annotators to work in.

The MPQA annotation scheme will be described only briefly here. The main perspective annotations include direct expressions of potential opinions (namely, "speech events" and "private states" —together referred to, somewhat obscurely, as "ons"), and indirect expressions of opinions (namely, "expressive subjectivity"). Other annotations may include the sources and targets of these opinion expressions, the strengths of the opinions, the polarity (negative or positive) of the opinions, and, for direct opinions, whether the opinion was presented factively or not.

As an example, consider (3):

(3) "It is [$_{ES}$ heresy]:' [$_{ON}$ said] Cao. "The 'Shouters' [$_{ON}$ claim] they are [$_{ES}$ bigger than] Jesus."

This example contains direct speech events *(ons)* by Cao and the 'Shouters'. In addition, there are expressions where Cao's opinions are expressed indirectly (ES),

including *heresy* and *bigger than*.

The annotation architecture was implemented using the annotation tool included in the GATE text processing framework (Cunningham *et al.* 2002). The annotation process is preceded by a document preparation phase. Annotators add perspective information to the document. When complete, these annotations are transferred to the annotation database.

To prepare documents for annotation, the raw text is extracted. Original markup (e.g., SGML markup for title, author, source, date, etc.) is moved to the annotation database. The document is imported into GATE and tokens, sentences, and part-of-speech tags are identified using components included with GATE**.** A number of annotations are automatically added to the document. Since each sentence is considered an "implicit" speech event of the writer, these annotations are added automatically. By default, they are factive, but the annotator may change this value.

When a document is completely annotated, the annotations are exported to the annotation database by a custom GATE component that we implemented. Another custom GATE component is available to verify a few correctness properties of the perspective annotations. For example, the checker will warn the annotator if there is an opinion associated with a source, but the source is not identified within the document.

Using the annotation architecture, we have annotated over 100 documents with perspective information. Moreover, the results of an agreement study are given in the Results section. The good results of the agreement study demonstrate that it is possible to annotate opinion information.

Learning Architecture

The learning architecture supports the development of components that learn to automatically identify perspective information in text. The goals of the learning architecture are:

- to facilitate the use of manually annotated documents as training input for the learning algorithms;
- to facilitate integration of a variety of text processing components as producers of features for the learning algorithms;
- to facilitate experimentation with various components and features within a flexible, modular framework.
- to facilitate evaluation of experimental results.

Both the *instances* and *features* employed in machine learning originate from the annotation database. Instances are represented as annotations, and feature values are represented as annotations that occur in the context of one of the instances, allowing both instances and features to be associated with portions of the document. The annotation database thus provides a single tool for managing all the information in the architecture.

A *feature generator* is a program that consumes a document and its annotations as input, and produces more annotations as output indicating the features detected in the document. An *instance generator* is a program that consumes a document and its annotations as input, and produces output corresponding to the instances of some machine learning task. For example, to learn to identify *ons* (direct expressions of opinion), an instance generator might collect all the verb groups of a document as potential ons, and one of the feature generators might annotate spans of quoted text in the document. Both instances and feature annotations may depend on other feature annotations. For example, the potential *on* generator above depends on parse annotations to indicate the existence of the verb groups. The suite of generator programs, coupled with the annotation representation, and the database, provides a flexible architecture for composing training data for learning. Feature generation and instance generation are discussed in more detail below.

Instance and feature annotations can be compiled together and converted to a form suitable for use as training data. In a preliminary experiment, we used this architecture to learn to automatically identify private states and speech events *(ons)*. The description and results of the experiments are reported in the Results section. To summarize the results, we trained two classifiers—using naive Bayes and k-nearest neighbor algorithms, both of which exceeded the performance of a heuristic baseline system. We currently achieve up to 66.4% f-measure for identifying ons.

The remainder of this section describes the features currently included in the learning architecture.

Text Processing The current implementation of the learning architecture includes a number of text processing components.

- GATE tokenization, sentence splitting, part-of-speech tagging. These preprocessing components are executed together within GATE.
- Alembic tokenization, sentence splitting, part-of-speech tagging. MITRE's Alembic components are an alternate source of token, sentence, and part-of-speech annotations.
- Stemmers. Stem annotations are available from both Porters and Abney's stemmers.
- CASS. CASS is a shallow parser that constructs a flat syntactic structure for the document, including noun and verb chunks, prepositional phrases, and clause chunks.

- Phrag. Phrag named entity annotations indicate the presence of entities such as persons, organizations, locations and dates.

Feature Processing In addition to text processing feature generators of the sort listed above, the architecture also facilitates a more declarative specification of features, with a corresponding feature generation program to locate and annotate features according to the specification.

The feature specification language, called TFF, encodes feature patterns over words. A pattern indicates the length of the feature in words and the particular words and part-of-speech tags that may occur. Additionally, the pattern also indicates the type of the resulting feature annotation. Pattern (4) is an example:

(4) type=fixed4gram len=4 word1=what pos1=pronoun stemmed1=y word2=a pos2=DT stemmed2=y word3=bunch pos3=noun stemmed3=y word4=of pos4=IN stemmed4=y

This pattern matches, for example. 'What a bunch of nonsense!"

The following is a current list of TFF feature specifications:

- Speech event verbs from Ballmer and Brennenstuhl (Ballmer & Brennenstuhl 1981). from Levin (Levin 1993), and from Framenet (Framenet).
- Psych verbs from Levin (Levin 1993) and from Framenet (Framenet).
- Potential subjective element words and phrases from Wiebe et al. (Wiebe *et al.* 2002).
- Subjective patterns induced via the meta-bootstrapping process (Thelen & Riloff 2002).

Application Architecture

The application architecture supports the perspective clustering task. The goals for the application architecture are:

- To establish a framework for exploring what aspects of opinions are likely to be the most useful for accomplishing opinion tasks that would be of direct interest to analyst users.
- To establish a framework for evaluating opinion tasks.
- To conduct an example evaluation to explore what obstacles will be faced in a full evaluation.

The architecture has three stages—document retrieval, perspective identification, and passage clustering. The document retrieval stage employs the SMART information retrieval system. In principle, the perspective identification stage employs the components trained within the learning architecture. However, for the evaluation reported in the Results section, perspective identification is actually performed by our heuristic baseline system (described in the Framework section), since the learning experiments and clustering experiments were occurring simultaneously.

For each document relevant to the query, SMART selects the best passage. Candidate passages are determined by a simple static algorithm that targets passages of length about 800 characters, broken on sentence boundaries. Overlapping passages are used so that the first passage might be the first 900 characters of a document (ending at the first sentence break after 800 characters), and the second candidate passage might start at character 425 and end at character 1300, again containing only complete sentences.

We implemented a two-phase agglomerative clustering approach to group the best passages. Initially, we start off with each passage in a cluster by itself and compute the similarity of every cluster to every other cluster by computing the passage-passage similarity. In the first phase, we perform a complete-link merging of clusters. We take the two clusters with highest similarity to each other and then merge them. Afterwards we compute the new similarity between the newly merged cluster, A, and each other cluster, B, by defining the cluster similarity to be the minimum passage-passage similarity between each passage of A and each passage of B. We then repeat the process of merging the two clusters with highest similarity, until that similarity is below some threshold. Thus, two clusters in phase 1 will be merged only if every passage in the first cluster has a sufficiently high similarity to every passage in the second cluster This is a very strict merging criteria meant to ensure the core clusters are very tight.

Group	ON Agreement	ES Agreement
1	0.8450	0.5031
2	0.7391	0.5034
3	0.8448	0.6895

Table 1: Interannotator agreement for *ons* and *expressive-subjective* elements

The second phase, invoked after no cluster-cluster complete-link similarity is above the threshold, is to perform an average-link merging of clusters. In this phase, the similarity between cluster A and cluster B is defined to be the average of the similarities of the passages in cluster A to those in cluster B. This is a much looser criteria and is appropriate for merging the tight clusters found in phase 1.

Clusters are merged in phase 2 until there are only 3 result clusters. There is an additional criteria that no cluster can contain more than 2/3 of the passages. This ensures that the result is not one huge cluster with 2 outlier passages forming their own clusters.

Results

Annotation Experiments

The purpose of the interannotator agreement study is to validate our annotations by assessing the consistency of human annotation. In pilot interannotator agreement experiments, we examined agreement for *on*s and *expressive-subjective* elements.

Three groups of annotators were involved in the study. Groups 1 and 2 each consisted of three project members. Group 3 consisted of a project member and a paid annotator. Within Groups 1 and 2, there was no prior training among annotators, in that no two of them had annotated the same documents and then discussed their results. However, the annotation instructions had been presented to them before, and each of them had annotated some documents. The annotators in Group 3 had trained together before. Each group annotated a set of three or four documents.

Annotators differ from one another concerning the boundaries of the ons and expressive-subjective elements they identify. For applications, it is probably most important that both annotators see an opinion expression within the same text span, and not that their exact boundaries match. In the experiments, we count overlapping ons and overlapping expressive subjective elements as matches.

Suppose a and *b* are two annotators. For measuring agreement on between a and *b,* we calculated agr (a||*b),* defined to be the proportion of as annotations that were found by *b.* This measure is appropriate considering that two annotators will not identify the same number of elements. Since agr (a||*b)* is directional, we also calculated *agr (b||a)* for each pair. The agreement for a group is the average of all pairwise agreement scores.

Table 1 presents interannotator agreement results. The results for annotating ons are particularly encouraging given that the team members did not train among themselves. The *expressive-subjective* results are lower. However, the pattern

Algorithm	Precision	Recall	F-measure
Baseline	69.9	47.7	56.7
Naïve Bayes	46.7	76.6	58.0
K-NN	69.6	63.4	66.4

Table 2: Performance results for tagging ons

of agreement among the annotators within a group is far from random. As it happens, in each of Groups 1 and 2 there is one particularly sensitive annotator who identifies many more *expressive-subjective* elements than the other two members of his or her group. It turns out that the other two members' annotations are largely subsets of the sensitive annotators' annotations. This is not necessarily surprising, because we did not calibrate the sensitivity of annotators' judgments of *expressive-subjective* elements. Indeed, for various applications, it is likely that either more or less sensitivity may be appropriate. This is a fruitful area for further investigation.

In addition to these agreement results, we achieved up to 80% kappa agreement on the *only-factive* task for *ons* that two annotators agreed upon and that had certain *only-factive* judgments.

Learning Experiments

The purpose of the learning experiments is to determine whether automatic taggers of perspective information can be trained using the annotated documents. Our initial experiments target automatic tagging of single-word direct opinion expressions (*ons*).

For baseline *on* identification, we use two lists of speech event verbs. If a word's lemma was found on one of the word lists, we tag it; other words and word-sequences are left unmarked. The two word lists come from Levin (Levin 1993) and Framenet (Framenet).

For learning *ons* we used the naive Bayes and k-nearest-neighbor implementations included with the Weka machine learning toolkit (Witten & Frank 1999). Each word in a training document comprised a training instance. Features included all words within 2 words on either side of the target word, the part of speech of the target word, the category from the same two word lists used in the baseline system. We also used some features derived from the CASS (Abney 1996) partial parser—the categories of the current word's chunk, of the previous chunk, and of the next chunk. For training, we used all the data annotated at the time we ran the experiment. The training data consisted of 92 annotated documents containing 63,586 potential *on* instances.

Performance is measured using recall, precision, and f--measure. Given sets of entities G and S annotated in the gold standard and by the system, respectively, we have recall R = |G∩S|/|G|, precision P = |G∩S|/|S|, and f-measure F = 2PR/(P+R).

Table 2 presents the results of the *on* tagging experiments. Results for naive Bayes and K-NN are averages over 10-fold cross-validation. We were pleased that by the F-measure statistic, both learning algorithms bested the

Zimbabwe								
Cluster	Base				Opinions			
	Both	Yes	No	Neither	Both	Yes	No	Neither

1	0	3	11	18	1	5	8	15
2	1	1	5	6	1	1	8	8
3	2	2	4	4	0	1	3	6
Kyoto								
Cluster	Base				Opinions			
	Both	Yes	No	Neither	Both	Yes	No	Neither
1	0	0	7	7	0	2	2	8
2	1	3	4	7	2	2	19	8
3	3	2	3	3	2	1	2	1

Table 3: Cluster evaluation

baseline. These results indicate the feasibility of learning perspective information.

Application Experiments

The purpose of the experiments involving the perspective clustering application is to determine whether perspective information could be useful in applications of interest to an information analyst.

Manual Clustering Study A first step in looking at automatically clustering documents is to examine how humans cluster, and what are the important issues for humans. Six MPQA project members plus an ex-analyst manually clustered opinions from documents related to 3 topics:

1. Election in Zimbabwe.
2. Treatment of prisoners at Camp X-Ray, Guantanamo Bay.
3. President Bush's alternative to the Kyoto Protocol.

There were 19-31 documents per topic, with multiple opinions per document. Since the purpose was to explore what humans might do, the instructions were deliberately vague.

As might be expected given the lack of instructions, the participant background strongly influenced the type of clusters. One project member, a linguist, separately clustered every sentence according to the perceived purpose of the sentence. This would be useful for information extraction to database. The ex-analyst clustered according to whether immediate threat of violence existed. Four people clustered roughly according to the proposed end-user task format: they separated opinions into pro/con top-level clusters, and then broke those down into sub-clusters. Nobody's sub-clusters or even sub-cluster strategy agreed with anybody else's.

Two major issues that came out of the discussion were the treatment of supporting evidence and how to handle outlier opinions that didn't match other opinions using whatever strategy was being used.

All participants agreed that treatment of supporting evidence was important, but they disagreed on how to include it. For example, one had a separate sub-clustering just for evidence. Some included evidence as part of an opinion, others did not. Everybody agreed there needed to be some way of linking evidence to opinion.

The major question of involving outliers was how could we distinguish random outliers from outliers that would be important to an analyst. People wanted several opinions in each of their clusters or sub-clusters, but an analyst will often be much more interested in the exceptions: in the one agent in a group whose opinion or tone does not match the rest of the group. No general solution to the problem of outliers was proposed, though it was noted that the particular situation with pro/con top-level cluster offers the ability to duplicate sub-clusters in both the pro and con clusters, thus an important exception might appear on the other side as a sub-cluster of size one. We measured agreement among the four pro/con two-level cluster participants. The overlap between the sets of "pro" opinions of two participants ranged from 50-80%. The numbers are a bit fuzzy since participants defined opinion boundaries differently. There was very weak agreement at the sub-cluster level, even if two participants constructed sub-clustered using the same basis. For example, even if the sub- clusters are formed using the type of agent expressing the opinion, participants differed as to whether the head of a government task force speaks for the government.

We also measured whether people agreed on the boundaries of opinion segments. In general, segment boundary agreement was about 60% for those participants who treated evidence the same way.

Overall, the lesson learned from this exploratory task is that clustering is demonstrably important and useful, but everybody does it differently for different reasons. This implies that any evaluation of clustering must be relative to a very clearly defined task. In addition, gold standard evaluation of clusters, where a system's clustering is compared against a pre-defined "correct" clustering, is going to be very difficult for anything other than a simple clustering task. Also, outlier evaluation must be explicitly addressed for those tasks where it is considered important, and it will not be easy.

Clustering Evaluation Our final experiments evaluate the end-user perspective clustering task.

We constructed a new collection of 271,822 foreign news documents from June, 2001 to May, 2002. The vast majority of these documents are from FBIS, Foreign Broadcast Information Service, with a very small number (157) of other documents gathered from the MITRE MITAP systems. (These extra documents were part of our pilot investigation done before settling on FBIS for the bulk of the collection.) The total size of the collection is about 1.6 GBytes.

We also constructed a set of 8 topic questions. All 8 topic questions are pro/con questions similar to question (1). We ran these topics using SMART with relevance feedback on the full FBIS collection. We identified 40-105 related documents per topic (not all relevant to the original topic).

For 4 of the 8 topics we manually identified segments in all the related documents that answered the pro/con question. There were generally 0 -- 4 answer segments per document, with each segment generally consisting of 1—3 sentences. There was an average of 1.1 answer segments per document. For each answer segment we store the agent expressing the answer, and the start and end of the segment.

For each of the four topics in the collection, we find the single best passage within each related document that answers the question. We then cluster these passages into a small number of clusters (3 was used here) and evaluate using the manually determined answer segments. The clustering is good if "like" opinions (either pro or con) occur together, as determined by the answer segments within each clustered passage.

The above process is performed twice. In the first experiment, the determination of best passage and the clustering between passages is dependent on the terms within the candidate passages only. In the second experiment, we boost the importance of the candidate passages and their related similarities if the passage contains an automatically determined *on* using the simple word list based heuristics described in the Framework section. We would hope that the second trial will contain more opinions (as determined by presence of answer segments), and that those passages would be better clustered into "like" opinions.

Table 3 gives the results for the Zimbabwe topic. For the Base trial, where passages were chosen and compared independent of opinions, the Yes, No and Neither answers in the answer segments were scattered pretty randomly throughout the 3 clusters. For the Opinions trial, where automatic detection of opinions was used to select and compare passages, the distribution of Yes/No answers among the 3 clusters improved a bit. Given the experimental design where clusters are forced to be merged, success occurs if the minority opinions (in this case Yes) are clustered together, possibly with some majority opinions added on. For this topic, 6 out of the 9 Yes opinions (including the Both figures) occur in one cluster. So this aspect of the results yielded a minor improvement.

The number of passages that contained no answer to the topic question remained just as large in the Opinions trial as in the Base trial. That's a clear-cut failure of our algorithms to incorporate opinions into the passage selection process.

Zimbabwe			
Cluster	Both	Yes	No
1	1	1	18
2	0	1	10
3	0	23	34
Kyoto			
Cluster	Both	Yes	No
1	1	6	19
2	0	2	1
3	0	4	7

Table 4: Retrospective cluster evaluation

Different passages were often chosen, but the passages sometimes included opinion indicators that were unrelated to the topic. This lack of coherence is a weakness of using static passages; this needs to be explored in future experiments.

The results of the Kyoto topic are given in Table 3. If anything, the results were less successful than the Zimbabwe topic. Once again, the number of passages without answer segments remained the same as opinion evidence was added. That result is more reasonable for this topic than for the Zimbabwe topic; most of the passages containing neither answer were in documents themselves that did not contain either answer (non-relevant documents). Given the experimental set-up, nothing can be done with those documents. The minority answer for this topic (again Yes) became a bit more spread out among the 3 clusters instead of less spread out. So this experimental result indicates a failure for our opinion algorithms for this topic also.

The two topics are fairly different when the type of opinions is looked at qualitatively. The Zimbabwe opinions tend to be rather crisp and short with substantiating factual evidence. The Kyoto opinions tend to be longer and not as strongly stated. Any kind of clustering or analysis of the Kyoto opinions will be less successful. Any future work in the area will need to ensure that enough topics of varying difficulty are included.

Retrospective Evaluation Was the poor performance of the sample simple evaluation task due to the difficulty in finding opinions, or to the clustering of these opinions?

Suppose we could find opinion passages perfectly. Would our algorithms then be able to cluster them well?

These questions suggest a simple retrospective evaluation: Take all passages containing the topic answers themselves (giving us perfect knowledge about relevant opinions). Cluster these passages using the same algorithms as previously.

Table 4 gives the results for the same Zimbabwe and Kyoto topic discussed above, except using the answer segments as passages. The Zimbabwe topic gives almost perfect results. Almost all of the Yes answers, 23 out of 26, occur in Cluster 3. There are a fair number of No answers in that cluster also, but that's unavoidable in this experimental design that forces clusters together.

The Kyoto topic is again a failure. We were not able to group the Yes answers into a single cluster.

There are several important differences in the type of passages being clustered in this retrospective experiment as opposed to the original simple experiment. For the Zimbabwe topic, the passages tended to be shorter and much more coherent. The Kyoto passages were fuzzier and longer than the Zimbabwe answers, sometimes including the entire document. This fuzziness undoubtedly contributed to the Kyoto clustering failures. In each case, there were multiple passages per document.

However, the important result here is not the actual clustering experiments, which were hastily done at the very end of the MPQA workshop, but the experimental design, which considerably more attention was paid to. We have given a reasonable end-user task involving opinions, and shown a method to evaluate the success (and failures) of our algorithms.

Conclusion

The MPQA project took a comprehensive look at using perspective information in question answering. In addition to formulating an evaluation methodology based on an end-user opinion clustering application, we executed a successful program of annotating opinion expressions in documents, and experimented with machine learning based automatic perspective taggers.

We developed an annotation scheme to represent information about perspectives in text, and we have annotated over 100 documents. Good agreement results indicate that opinion annotation is a tractable task, and suggest future directions for improving the annotations.

Using the annotated documents as training data, we trained a classifier to recognize single-word *ons*. The success of this classifier indicates that corpus-based learning of perspective information is a feasible endeavor.

We designed an end-user yes/no clustering application that facilitates evaluation of the utility of perspective information in question answering. In preliminary clustering experiments, we found that opinion information sometimes produces better clusters. More importantly, we verified that our evaluation methodology detects success and failure in our application.

All of the work reported here is ongoing. Annotation of perspective information continues, and further agreement studies are planned. We also plan to continue experiments in learning to identify perspective in text by adding *expressive-subjective* and *only-factive* tagging tasks. As the experiments proceed we hope to identify linguistic features that help to classify opinions. Finally, we plan to improve the clustering application by using more training data, improving the automatic tagging of opinions, and improving the clustering algorithms. Ultimately, as we begin to understand the role of perspective in question answering, we hope to move on to other question answering tasks that incorporate perspective more fully into answers.

References

Abney, S. 1996. Partial parsing via finite-state cascades. *J. of Natural Language Engineering* 2(4):337—344.

Ballmer, T., and Brennenstuhl, W. 1981. *Speech Act Classification: A Study in the Lexical Analysis of English Speech Activity Verbs.* Springer-Verlag.

Cunningham, H.; Maynard, D.; Bontcheva, K.; and Tablan, V. 2002. GATE: A framework and graphical development environment for robust nlp toos and applications. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics.*

Framenet. See http://www.icsi.berkeley.edu/~framenet/.

Levin, B. 1993. *English Verb Classes and Alternations: A Preliminary Investigation.* Chicago: University of Chicago Press.

Thelen, M., and Riloff, E. 2002. A Bootstrapping Method for Learning Semantic Lexicons Using Extraction Pattern Contexts. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing.*

Wiebe, J.; Wilson, T.; Bruce, R.; Bell, M.; and Martin, M. 2002a. Learning subjective language. Computer science technical report tr-02-100, University of Pittsburgh.

Witten, I.H., and Frank, E. 1999. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan Kaufmann.