NRRC Summer Workshop on Multiple-Perspective Question Answering Final Report

Janyce Wiebe

Eric Breck, Chris Buckley, Claire Cardie, Paul Davis
Bruce Fraser, Diane Litman, David Pierce, Ellen Riloff
Theresa Wilson

Contents

Chapter 1

Executive Summary

1.1 Introduction

A summer workshop was held on the topic of Multi-Perspective Question Answering. The workshop was funded by the Northeast Regional Reseach Center (NRRC) which is sponsored by the Advanced Research and Development Activity (ARDA).

A group of resarchers and PhD students worked together to explore the area of Multi-Perspective Question Answering (MPQA). The accomplishments include a knowledge representation scheme to support manual annotation and analysis of data; a repository of linguistic clues relevant for perspective; a data corpus; a set of manually annotated data; an annotation system to support manual annotation; an application architecture; and the results of various types of evaluation.

The problem we address is finding and organizing expressions of opinions in the world press and other text. Our 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, clustering them and their sources in various 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.

We focused on building a comprehensive infrastructure to support exploration of this area, rather than focusing on one particular piece in depth. Thus, we built and evaluated end-to-end systems, and performed both deep and high-level annotations of the data. In addition, we developed a representation and language for describing how opinions are expressed in language, which provides a firm basis and which is expressive and extendable.

The remainder of this executive summary overviews the activities and accomplishments of our research effort. Chapter 2 of this report provides greater detail. Chapter 3 is a catalogue of results and technical products of our work.

1.2   Overview of Activities and Accomplishments

To support high-level tasks, such as building perspective profiles over time, and recognizing trends and significant changes in opinions, we developed a language and representation to describe basic building blocks of how opinions are expressed in language. Our work was informed by work in other fields, such as linguistics and literary theory. But our requirements for this project go beyond descriptive linguistics work. The need to support computational work raises additional demands. First, we must address the ambiguities that arise in text. Second, we must identify a set of concepts and properties that are not overly detailed, but are rich enough to capture needed information. Third, annotators must be able to reliably and consistently manually annotate data, so that high-quality training and test data may be developed.

Thus, we developed annotation instructions for identifying expressions of opinions in text. A knowledge representation scheme was developed, and implemented in a system that supports manual annotation of the data. A conceptualization was written that fleshes out concepts used in the annotation instructions. Over 100 documents have been annotated according to the instructions. Pilot agreement studies were performed, with encouraging results. Three people were trained as annotators. An annotation agreement study was performed with two of the trained annotators, showing high agreement. We are eager to continue our work performing and analyzing annotations of opinions.

A framework and initial design were developed for defining summary representations of opinions, building on the annotations described above. The goal is to provide concise and ultimately user-tailored summaries of the opinions expressed in an article, in a set of articles, or in any arbitrary segment of text. Desirable facilities are support of direct querying, collective perspectives, matching user-specified constraints, creating perspective profiles, debugging, and creating a higher-level gold-standard for evaluating Natural Language Processing Systems. Working toward implementation and experimentation with this

4

framework, building on our work on manual and automatic annotation, is a promising area of research.

A repository of linguistic clues was created that are promising indicators of perspective. It includes words and phrases from existing published work, and words, phrases, and patterns learned using automatic processes.

An overall solution architecture was designed and implemented. It includes a manual-annotation architecture, a learning architecture, and an application architecture.

The manual annotation architecture includes pre-processing components, such as sentence splitting, an annotation system which enables human annotators to annotate the data (implemented using Sheffield's Gate system), and post-processing components to produce data that may be passed as input to the learning and application architectures.

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

- to facilitate the use of MPQA 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.

We performed several initial experiments to reproduce some of the manual annotations automatically, using the learning architecture. Although these experiments are preliminary, significant improvements over baseline accuracy were achieved for a major part of the annotations. We are currently planning additional experiments to perform in future research, which will target additional aspects of the annotation scheme, will involve more of the features from the repository of linguistic clues, and will involve additional learning algorithms, such as co-training.

The final component of the MPQA workshop is the End-User Evaluation. There are three main goals of the End-User Evaluation. First, we wanted to explore what aspects of opinions are likely to be the most useful for accomplishing opinion tasks that would be of direct interest to analyst users. Next, we wanted to establish a framework for evaluating opinion tasks. Finally, we wanted to conduct an example evaluation to explore what obstacles will be faced in a full evaluation.

Two two-hour presentations were prepared for the midterm and final meetings.

Pre-workshop planning and development of initial annotation instructions and annotation system were performed during Spring 2002.

The two months after the workshop has also seen a high level of activity, including revision of annotation instructions, annotation of additional documents, designof an inter-annotator agreement study, preparation of final report, and planning of machine learning experiments.

The infrastructure and data for this work are at MITRE. There is no current support to maintain that data and infrastructure, to migrate them to team members' home institutions, or to support team member travel to plan experiments. There is a concern that the infrastructure and collaborations will erode without short-term funding for the above.

Chapter 2

Description of Workshop Activities and Technical Results

2.1   Participants and Conduct of the Workshop

A total of 10 people were involved in the workshop. Some received compensation for two months of work (full-time), some for one month of work (half-time), and one person (Theresa Wilson) also received compensation for the spring 2002 semester as a research assistant. Many people devoted significantly more time to the effort. Wiebe is currently supporting Wilson to work on this project, using research funds from another source.

The participants are:

- Chris Buckley, President of SabIR Research Inc., full-time

- Eric Breck, Graduate Student at Cornell University, full-time

- Claire Cardie, Professor at Cornell University, half-time

- Paul Davis, Graduate Student (who recently finished his PhD) at the Ohio State University, full-time

- Bruce Fraser, Professor at Boston University, half-time

- Diane Litman, Professor at University of Pittsburgh, half-time

- Ellen Riloff, Professor at University of Utah, half-time

- David Pierce, Professor at SUNY at Buffalo, full-time

- Janyce Wiebe, Professor at University of Pittsburgh, full-time

- Theresa Wilson, Graduate Student at University of Pittsburgh, full-time

The schedule was the following:

- March 13-15, 2002: A kickoff meeting was held at the University of Pittsburgh. Seven of the workshop participants (Breck, Buckley, Cardie, Litman, Pierce, Wiebe, and Wilson), as well as Kelcy Allwein, David Day, Penny Lehtola, Mark Maybury and John Prange attended.

- April 1-2, 2002: Fraser and Wiebe met at the University of Pittsburgh to plan work for the summer.

- April 11, 2002: Cardie, Wiebe and Wilson met at the University of Pittsburgh to work on the initial manual annotation system.

- May 20-July 23: The workshop itself. The following special meetings were held during this time:

  - June 6: Midterm Meeting and Presentations
  - June 7: Intelligence Speaker Series
  - June 11-13: AQUAINT PI Midterm Meeting (Wiebe attended)
  - July 22-23: Final meeting and Presentations

- August and September: Annotation instructions revised and extended, additional documents annotated, machine learning experiments planned, final report prepared.

2.2 Data Acquisition and Corpus Formation

The collection of data gathered for this project is a large collection of over 270,000 documents that appeared in the world press over an 11-month period, from June 2001 to May 2002. The documents were downloaded from the MITRE MiTAP system. With the exception of a small number of relevant documents, all documents in the collection are documents taken from FBIS.

The FBIS document collection has the following characteristics. It is an English language collection, with 60% of the documents translated into English by FBIS. 20% of the documents are transcriptions from television or radio broadcasts. 5% of the documents are explicitly identified as editorials.

Using an information retrieval system, eight topics were used to select a subset of the full collection. The topics are: the presidential election in Zimbabwe, the U.S. annual human rights report, relations between Taiwan and China, the U.S. holding of al Quaida

and Taliban detainees at Guantanamo Bay, passage of the Kyoto Protocol, the political upheaval in Venezuela, Israeli settlements in the West Bank, and reactions to the U.S. characterization of certain countries as an "axis of evil."

At least 200 documents were retrieved on each of these topics. Of this set, 575 documents were identified as being publicly available for a small fee from the World News Connection (WNC), a division of the federal National Technical Information Service (NTIS). With the help of MITRE, plans are currently being made to have WNC distribute both the data and the annotations.

2.3   Annotation of Expressions of Opinions in Language

2.3.1   Instructions for Annotating Opinions in Newspaper Articles

The *Instructions for Annotating Opinions in Newspaper Articles* provide a conceptualization framework for the annotation task. Briefly, the annotations are centered around two main types of things: (1) explicitly mentioned private states and speech events, (e.g., "John hates Bill" and "Mary said she would be home late"), and (2) expressive subjective elements (e.g., "to put it mildly" and "what an idiot"). The annotation task involves identifying text spans that correspond to these concepts, as well as related information, including the source (e.g., whose opinion is being expressed?), type (e.g., is the opinion a positive one?), and strength of the private state (e.g., medium-strength "criticize" versus extreme-strength "vehemently attack").

We use the term, "on", as a shorthand reference for the word or phrase that explicitly mentions a private state or speech event. If an opinion or other private state is being expressed by a source, the instructions direct the annotator to classify the on for that source as onlyfactive=no. Otherwise, the on is tagged as onlyfactive=yes.

2.3.2   Annotating Opinions in Newspaper Articles: Example Passages with Annotations

To assist the annotators, we created a reference document, *Annotating Opinions in Newspaper Articles: Example Passages with Annotations*. This document explains the annotations for a number of difficult, real-world examples that were encountered during annotator training. The examples in this document also are intended to help the annotator by making more concrete the boundry between when and when not to annotate a private state or speech event.

2.3.3 Opinion Annotation in GATE

The annotation system that we used for the workshop was developed using GATE, a General Architecture for Text Engineering. GATE is freely available from the University of Sheffield, http://gate.ac.uk.

We specified the types of annotations described in the *Instructions for Annotating Opinions in Newspaper Articles* using XML. The XML schemas that specify the annotation types are located at http://www.cs.pitt.edu/mpqa/opinion-annotations/gate-annotation. GATE with our XML annotation schema becomes a customized annotation tool. We developed online instructions in html for using GATE to annotate opinions, which include the pointers to Gate and the XML schemas mentioned above.

2.3.4 Annotation Sample

This section provides a brief example of the opinion annotations.

The following is the first sentence from an article about the 2002 presidential election in Zimbabwe. The article appeared on March 15, 2002 in the newspaper, *Dawn*.

> Western countries were left frustrated and impotent after Robert Mugabe formally declared that he had overwhelmingly won Zimbabwe's presidential election.

There are three agents that are sources in this sentence: (1) the writer, (2) Western countries, and (3) Robert Mugabe. Here are their annotations.
(1) writer

> on=implicit, onlyfactive=yes

There is no word or phrase that is the on for the writer, but everything written in the article is attributed to the writer, and must be evaluated. In this sentence, because there is no expressive subjectivity attributed to the writer, the implicit on for the writer is onlyfactive=yes.
(2) Western countries

> on=were left frustrated, onlyfactive=no, strength=medium

The on for Western countries is directly expressing an emotion This direct expression of a private state makes onlyfactive=no.
(3) Robert Mugabe

> on=formally declared, onlyfactive=no, strength=neutral

expressive-subjectivity=overwhelmingly, strength=medium

The on for Robert Mugabe is a speech event. The strength of the on is neutral because there is not a private state being expressed by the on-phrase. However, because of the expressive-subjective element, overwhelmingly, the on for Robert Mugabe is onlyfactive=no.

Figure 2.3.4 shows how these annotations look inside the GATE annotation tool.

More annotated examples can be found with the online GATE annotation instructions, http://www.cs.pitt.edu/mpqa/opinion-annotations/gate-instructions.
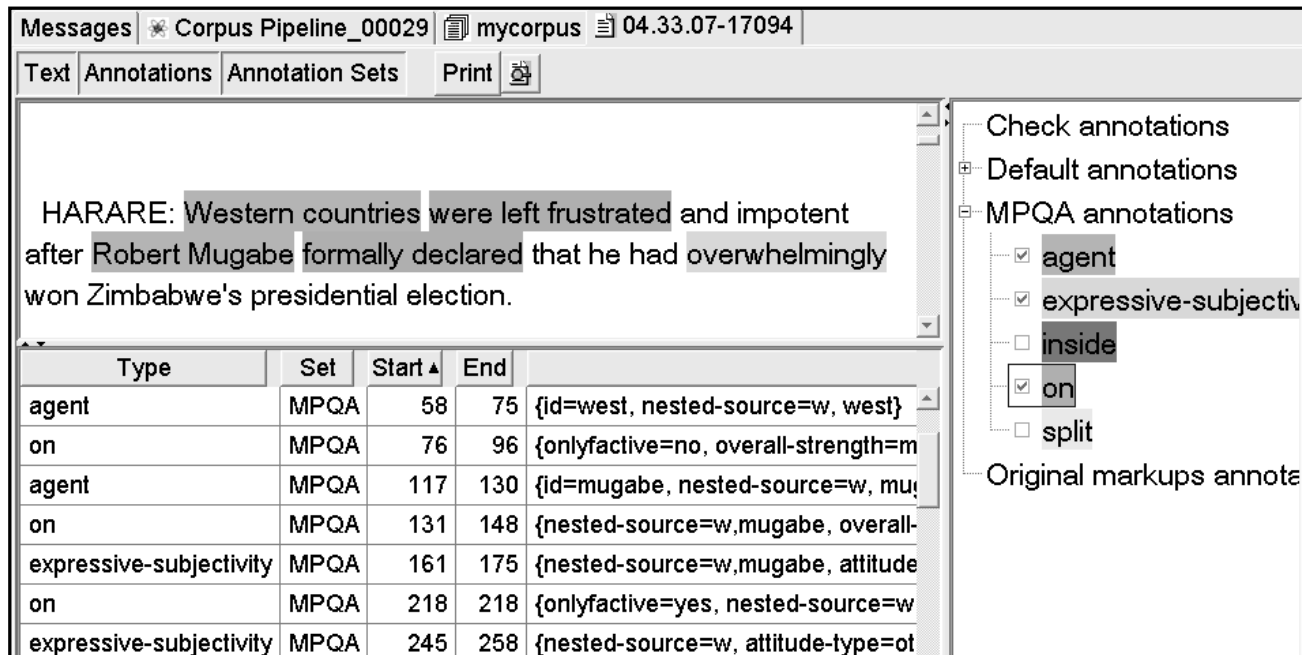
Figure 2.1: Example of annotations in GATE

2.3.5 Annotated Documents

As of September 22, 2002, a total of 114 documents have been annotated. 57 of these documents have full, deep annotations; the remaining documents have shallow annotations. With deep annotations, all relevant features are included for each text span identified by the annotator. This includes judgments of certainty. For shallow annotations, the annotator is asked to identify the pertinent text spans, as with the deep annotations, but is allowed

to leave the annotation features unspecified, with the exception of the onlyfactive feature. Although it captures less detail, performing shallow annotations on a document requires significantly less time than performing deep annotations: approximately 40 minutes, on average, compared to an average of 2 1/2 hours for for deep annotations. In addition, any documents with only shallow annotation can later be annotated in depth, building on the with the existing shallow annotations. That is, the shallow annotators are a subset of the deep annotations.

One goal of the workshop was to produce an annotated corpus of opinions that could be made available to other researchers. As it currently stands, it will be possible to license 28 (22 deep) of the annotated documents from the World News Consortium (WNC). All documents annotated in the future will also be drawn from the subset of those available from the WNC.

2.3.6 Interannotator Agreement

In order to validate the annotations we have defined, we need to assess the consistency of human annotation. To that end, we conducted pilot interannotator agreement experiments. In this preliminary study, we examined agreement for three aspects of the annotations: *on*s, *expressive-subjective* elements, and judgments of *only-factivity*.

There were three groups of annotators involved in the study. Group 1 consisted of 3 workshop participants. Group 2 consisted of 3 different workshop participants. And Group 3 consisted of 1 workshop participant and 1 paid annotator (who did not attend the workshop, and who resides in Pittsburgh). 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.

Group 1 annotated a set of 4 documents; Group 2 annotated a different set of 4 documents; and Group 3 annotated a different set of 3 docments. Below, we report the pairwise agreements for each pair within a group, for each of the three tasks. In addition to annotating *on*s, *expressive-subjective* elements, and judgments of *only-factivity*, the two annotators in Group 3 also indicated when their judgments were uncertain.

Note that annotators differ from one another concerning the boundaries of the ons and expressive-subjective elements they identify. For example, following is a sentence fragment in which two annotators identify different boundaries of an **on** (in boldface):

Bush has **adopted** the most pro-Taiwan posture of any president...

Bush has **adopted the most pro-Taiwan posture** of any president...

Following is an example with expressive subjective elements. The first annotator iden-
tifies only "alarming" as the expressive subjective element, while the second includes the
words before an after:

...some of Mr. Chavez's more **alarming** faults...

...some of Mr. Chavez's **more alarming faults**...

For applications, it is probably most important that both annotators see an expressive
subjective element or "on" within the same text span, and not that their exact boundaries
match. In the instructions, we did not attempt to define rules to try to enforce boundary
agreement. While such rules would likely be complex, they might be possible to define.
We suggest that a strong motivation for doing so is advisable before addressing this. In
the experiments, we count overlapping ons and overlapping expressive subjective elements
as matches.

Metrics

First consider measuring agreement on expressive subjective elements and ons. Two an-
notators will not, in general, identify the same number of objects: one will see more ons
than the other, or will see more expressive subjective elements than the other. The *agr*
agreement metric is an appropriate one for measuring agreement in this situation. This is
a directional measure of agreement.

Call two annotators a and b, and the sets of entities annotated by each A and B. We
compute the agreement of b to a as:

$$agr(a||b) = \frac{|A \cap B|}{|A|}$$

Note that this corresponds to the notion of precision and recall as used to evaluate,
for example, named entity evaluation. Our $agr(a||b)$ corresponds to the recall if a if the
gold-standard and b the system, and to precision, if they are reversed.

We now turn to measuring agreement for the only factive judgment. For two annotators,
a and b, we take the set of ons which they both identified, and calculate their agreement for
those. In this case, there are the same number of objects per annotator. Cohn's **Kappa**
(κ) (Cohen, 1960) metric is an appropriate measure of agreement in this case.[1] Let n_{ij} be
the number of judgments of an object i to category j, where in this case objects are ons, and
the available categories are onlyfactive-yes or onlyfactive-no. Let N be the total number

---

[1]This explication of κ is slightly abridged from (Wiebe et al., 1998), specialized for this two-judge,
binary judgment case.

of objects. Let $p_j = \frac{\Sigma_i n_{ij}}{2N}$, the percentage of assignments to category j. Let $Pe = \Sigma_j p_j^2$. Let $Pa = \frac{\Sigma_{i,j} \text{ where } n_{ij}=2 \ 1}{N}$, the fraction of objects that the judges agree on. $\kappa = \frac{Pa - Pe}{1 - Pe}$.

Results

a	b	$agr(b\|\|a)$	$agr(a\|\|b)$	average
chrisb	pcdavis	0.8706	0.8222	
chrisb	drpierce	0.8191	0.8556	
drpierce	pcdavis	0.8941	0.8085	
				0.8450
djl	tw	0.6607	0.8085	
djl	ebreck	0.8095	0.7447	
ebreck	tw	0.6250	0.8333	
				0.7391
anna	tw	0.8261	0.8636	0.8448

Table 2.1: Interannotator Agreement: ons

Table 2.1 presents the results for the interannotator agreement for marking ons, and Table 2.2 for expressive subjective elements[2]. The results for annotating ons are particularly encouraging given that the team members did not train among themselves.

The expressive subjectivity results are lower. However, the pattern 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. First, consider Group 1. T identified 153 expressive subjective elements in her groups' documents. E identified only 29, but fully 97% of those are included in T's set. Over 80% of Di's 76 expressive subjective elements are included in T's set. In the other group, Group 2, P identified 196 expressive subjective elements, D identified 75, and C identified 74. 88% of C's and 81% of B's are included in P's set. For various applications, it is likely that more (P and T) or less (D, C, E, and Di) sensitivity may be appropriate. This is another fruitful area for further investigation.

---

[2]The averages in these tables are arithmetic means of all the *agr*s. While *agr* corresponds to precision and recall, we feel that an arithmetic mean is a better way to average these results than the F-measure (harmonic mean) typical in, for example, named entity evaluation.

a	b	$agr(b\|a)$	$agr(a\|b)$	average
chrisb	pcdavis	0.3469	0.8784	
chrisb	drpierce	0.3600	0.3243	
drpierce	pcdavis	0.2959	0.8133	
				0.5031
djl	tw	0.3137	0.6184	
djl	ebreck	0.6897	0.2632	
ebreck	tw	0.1699	0.9655	
				0.5034
anna	tw	0.6138	0.7652	0.6895

Table 2.2: Interannotator Agreement: expressive subjective elements

Table 2.3 presents results for agreement on marking onlyfactivity judgments. Most interesting is the agreement between the trained annotator (anna) and one of our team members (tw). While the initial result is reasonably high, when the judgment set is reduced to only the subset where both annotators were certain of their judgment, which is fully 82% of the judgments, the agreement jumps to a Kappa value of 0.805. This is considered a very high Kappa value, as Kappa measures the amount of agreement over and above the agreement one would expect from chance.

2.4 Summary Representations of Opinions

Previous sections of this report described the linguistic annotation scheme that was designed as part of this project to support a wide variety of end-to-end applications in multi-perspective question answering. For any particular MPQA application, however, we anticipate the need to go beyond the low-level annotations and have begun to investigate the creation of **summary representations of opinions**, which would provide concise, and ultimately user-tailored summaries of the opinions expressed in an article, in a set of articles, or in any arbitrary segment of text.

In the subsections below, we first provide a concrete example of an MPQA summary representation for a portion of one article in the MPQA collection (section 2.4.1). We then briefly discuss how summary representations might be used in various MPQA tasks (section 2.4.2) and describe issues for the automatic creation of summary representations (section 2.4.3).

ann$_0$	ann$_1$	kappa
chrisb	pcdavis	0.5801
chrisb	drpierce	0.4485
drpierce	pcdavis	0.6518
djl	tw	0.4456
djl	ebreck	0.3248
ebreck	tw	0.5522
anna	tw	0.624
anna	tw [certain]	0.805

Table 2.3: interannotator agreement, onlyfactivity

2.4.1 An Example

An MPQA summary representation effectively encodes the gist of the opinions expressed throughout one or more texts or text spans. They are "summaries" in that they merge and make inferences from lower-level MPQA annotations that have been identified in the text.

As an example, consider the text in Figure /refsummary-rep-example, which is the first ten sentences of one document (#20.20.10-3414) from the Human Rights portion of the MPQA collection. The first sentence of the document,

> The Annual Human Rights Report of the US State Department has been strongly criticized and condemned by many countries.

should produce the following lower-level MPQA annotations:

<div align="center">

writer: *onlyfactive*

writer: *expressive-subj (medium).*

</div>

In particular, from the writer's perspective, the sentence can be classified as *onlyfactive*. In addition, the lexical cue "stongly" indicates some (medium) amount of expressive subjectivity.

A similar analysis of the remainder of the text fragment would produce the low-level annotations of Figure 2.3. It should be clear that the representation of opinions at this level is difficult for humans to absorb. It does, however, directly support the creation of an MPQA summary representation that provides the gist of the opinions expressed in the text. The MPQA summary representation for the sample text is shown in Figure 2.4. The summary makes it clear that there are three primary opinion-expressing agents in the text, at least according to the writer of the document — the writer him/herself; many

The Annual Human Rights Report of the US State Department has been strongly criticized and condemned by many countries. Though the report has been made public for 10 days, its contents, which are inaccurate and lacking good will, continue to be commented on by the world media.

Many countries in Asia, Europe, Africa, and Latin America have rejected the content of the US Human Rights Report, calling it a brazen distortion of the situation, a wrongful and illegitimate move, and an interference in the internal affairs of other countries.

Recently, the Information Office of the Chinese People's Congress released a report on human rights in the United States in 2001, criticizing violations of human rights there. The report quoting data from the Christian Science Monitor, points out that the murder rate in the United States is 5.5 per 100,000 people. In the United States, torture and pressure to confess crime is common. Many people have been sentenced to death for crime they did not commit as a result of an unjust legal system. More than 12 million children are living below the poverty line. According to the report, one American woman is beaten every 15 seconds. Evidence show that human rights violations in the United States have been ignored for many years.

Figure 2.2: MPQA Sample Text. First ten sentences from document #20.20.10-3414 from the Human Rights portion of the MPQA collection.

<writer>: onlyfactive   <writer>: expr-subj (medium)

<writer>: neg-attitude (medium) → <report> <writer>: neg-attitude (medium)
<writer>: neg-attitude (medium)

<writer>: onlyfactive   <writer, many-countries>: neg-attitude (medium)
→ <report>   <writer, many-countries>: extreme
<writer, many-countries>: neg-attitude (high, high, medium)

<writer>: onlyfactive
<writer, info-office>: neg-attitude (medium) → <US>
        <writer>: onlyfactive   <writer, chinarep>: onlyfactive
                                                <writer>: ?neg-attitude
(medium) → <US>   <writer>: expr-subj (low)   <writer>: neg-attitude (low)
→ <US>   <writer>: expr-subj (low) <writer>: neg-attitude (medium)
              <writer>: onlyfactive
                    <writer>: onlyfactive
<writer>: neg-attitude (low) → <US>   <writer>: expr-subj (low)

Figure 2.3: Set of Lower-Level MPQA Annotations for the Text Sample from Document
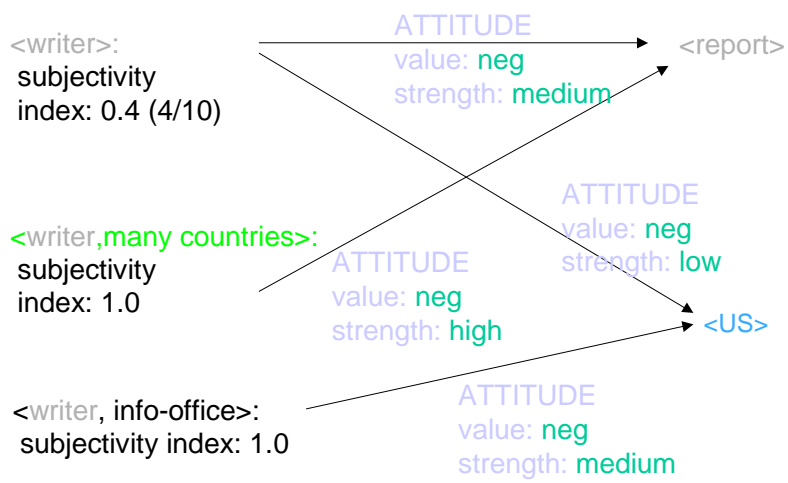#20.20.10-3414.

Figure 2.4: MPQA Summary Representation for the Text Sample from Document #20.20.10-3414.

countries in Asia, Europe, Latin America, and Africa; and the Chinese information office. Furthermore, these agents expressed the following opinions:

- the writer expressed a negative attitude (of medium strength) towards the human rights report;

- the writer also expressed a mildly negative attitude towards the United States;

- according to the writer, many countries (in Asia, Europe, Latin America, and Africa) expressed a strongly negative attitude towards the human rights report; and

- according to the writer, the Chinese information office expressed a negative attitude (of medium strength) towards the United States.

Inferences. As noted above, portions of the summary representation require making inferences across related groups of lower-level annotations. Associating a strength (low, medium, high) with each negative attitude is one such example. The *subjectivity index* associated with each nested agent is another example of the kind of summary statistic that one could generate from the lower-level annotations. It indicates, for example, that 4 out of 10 sentences of the writer include subjective language; and that all "utterances" associated with many countries and the Chinese information office include subjective content.

Internal representation. Like the lower-level MPQA annotations, the summary representation for a document, a set of documents, or one or more text fragments can be encoded as in-line annotations. This would allow for querying directly by the end-user.

Flexibility. There are many user-specified options for the level at which the MPQA summary representation could be generated. For example, the user might want summaries that focus only on particular agents, particular classes of agents, particular attitude types or attitude strengts. The user might also want to specify a particular level of nested source to include, e.g. create the summary from the point of view of only on the most nested sources.

2.4.2 Uses for Summary Representations

Although a primary use of summary representations is to provide a short, optionally tailored summary of the opinions expressed in a specific text(s) or text fragment, we anticipate other uses for the MPQA summary representations.

Direct querying. When the summary representation is stored as a set of document annotations, it can be directly queried directly by the end-user using XML "grep" utilities.

Collective perspectives. The summary representations can be used to describe the collective perspective w.r.t. some issue or object presented in an individual article, or across a set of articles.

User-specified views. The summary representations can be tailored to match (some types of) user-specified constraints, e.g. to describe the perspective of a particular writer, individual, government, or news service w.r.t. a particular issue or object in an individual article, across a set of articles.

Perspective Profiles. The MPQA summary representation would be the basis for creating a perspective "profile" for specific agents, groups, news sources, etc. The profiles, in turn, would serve as the basis for detecting changes in the opinion of agents, groups, countries, etc. over time.

Debugging. Because the summary representation is more readable than the lower level annotations, summary representations can be used to aid debugging of the lower-level annotations on which they were based. This is the case whether the lower-level annotations were manually generated or automatically generated.

Gold Standard "answer keys". Creating the "gold standard" by which to evaluate most empirical NLP tasks is generally an intensely time-consuming endeavor. Consider, for example, the amount of effort required to create the scenario template "answer keys" for information extraction evaluations. Once the gold standard for the lower-level annotations has been created for a collection, however, it may be possible to completely automate the creation of gold standards for various MPQA summary representations. These can then be used to evaluate summary representations created on top of *automatically* generated lower-level annotations.

Closer to true MPQA. The MPQA summary representations should let us get closer to true question-answering for multi-perspective questions. To handle TREC-style, short-answer questions, for example, a standard QA system strategy is to first map each natural language question into a question type (e.g. a "who" question, a "where" question, a "why" question) so that the appropriate class of answer (e.g. a person, a place) can be located in the collection. The MPQA summary representation acts as a question-answering template, defining the multi-perspective question types could be answered by our system.

2.4.3 Automatic Creation of Summary Representations of Opinion

In the paragraphs below, we discuss some of the issues involved in the automatic creation of MPQA summary representations.

Perfect lower-level annotations. Given a complete and accurate set of **deep** lower-level MPQA annotations, building a summary representation will be fairly easy, but still non-trivial. In particular, it requires accurate noun phrase coreference subsystem to identify the objects towards which some opinion was expressed. Although the deep annotations contain enough information to find all references to individual agents in a text, there is no plan to include similar annotations for objects. Identifying the object of an *attitude-towards* relation is often very difficult for human readers to determine and it is often not explicitly expressed in the text.

There are also likely to be complications when the text includes conflicting opinions.

Imperfect lower-level annotations. The situation becomes much harder, of course, when the MPQA summary representation is to be built on top of automatically generated lower-level annotations, which are likely to be incomplete and inaccurate. The situation will be akin to the information extraction task of "merging" extracted template relations into a scenario template. In addition, the noun phrase coreference system will be much more important — it will need to provide not only the links between coreferent objects, but also between coreferent agents.

Cross-document coreference. In contrast to the TREC-style QA task, effective MPQA will require collation of information across documents since the summary representation may span multiple documents. For example, if a user wants to know the range of perspectives on topic X, then the system will need to perform cross-document coreference w.r.t. topic X as well as w.r.t. the various agents that express views on the topic.

2.5 Proposal for Segmentation and Perspective Coherence

In the natural language processing literature, the term segmentation refers to breaking up a document into smaller chunks - or segments - that are locally coherent. Depending on factors such as corpus type and application need, different notions of coherence have been

proposed as the basis of segmentation.[3]

In the area of information retrieval, text segmentation has usually been based on semantic coherence. Segmentation is performed by placing segment boundaries at points of semantic discontinuity, which in turn are computed using measures such as lexical cohesion (Morris and Hirst, 1991; Hearst, 1997). In domains where large documents cover many topics, once a document has been segmented, the segments enable retrieval at the segment rather than the document level, or help guide the user to a portion of a retrieved full-document.

In the area of discourse analysis, segmentation has instead been based on notions of informational (Hobbs, ; Mann and Thompson, 1988), and/or intentional coherence (Grosz and Sidner, 1986; Passonneau and Litman, 1993). Determining which sentences are informationally coherent has often been computed using formal methods of inference (e.g. abduction) to prove that coherence relations (e.g., elaboration) relate the content or the information being conveyed within a segment. Segment boundaries have also been proposed based on analysis of discourse-level linguistic cohesive devices such as discourse markers and referring expressions. In contrast, intentional coherence has typically been based on a goal-oriented view of natural language processing; sentences are coherent when they can be related to the same purpose. Like informational coherence, intentional coherence can be computed using inference and/or linguistic clues. While informational and intentional coherence have typically been applied to monologues and dialogues, respectively, hybrid approaches have also been proposed.

For multi-perspective question answering, we believe that a notion of segmentation based on a new notion of "perspective coherence" will prove similarly useful for a variety of high-level tasks. In particular, in the next phase of our research, we would like to extend our annotations to include "perspective segments", by identifying sentence spans expressing coherent perspectives, which will be defined in terms of our existing sentence-level annotations. In other words, we believe that we can use our lower-level analysis to produce higher-levels of understanding, by looking at how sentences expressing perspective interact with one another in larger pieces of text. As with other notions of segmentation, this will likely involve merging and performing shallow inferences across sentences.

To motivate this idea, consider an example segment produced during an informal manual clustering study we performed, described below in section 2.10. For this study, workshop participants were asked to label opinions, where each opinion could be described by a single sentence, or by a segment consisting of a sentence span.

The excerpt in Figure 2.5 illustrates a sample segmentation from our coding exercise. In particular, 4 out of 7 coders placed sentences 3-8 through 3-10 in the same segment; a 5th coder placed the beginning of this segment one sentence earlier. The (deep) annotations,

---

[3]While many theories of segmentation are hierarchical and involving structuring the segments, for ease of explanation, we will focus here on the simpler case of linear segmentation.

3-6 Mugabe described the opposition as "donkey being controlled by the British," the former colonial power. *(on=described, only-factive=n, source=w,mugabe)*

SEGMENT BEGIN (1 CODER)

3-7 The fledgling MDC won 57 of 120 elected seats in June 2000 parliamentary elections as Mugabe's popularity plunged amid economic devastation and chaos.

SEGMENT BEGIN (4 CODERS)

3-8 The U.S. State Department released a human rights report on Zimbabwe Monday that accused the government of extrajudicial killings, undermining the independence of the judiciary and waging a "systematic campaign of violence targeting supporters and potential supporters of the opposition." *(on=accused, only-factive=n, source=w,us report)*

3-9 Security forces tortured opponents, ruling party militants abducted people, and police arrested opposition supporters who were themselves the victims of crimes. ****3-10**** Freedom of the press and freedom of assembly were also severely restricted, the report said. *(on=said, only-factive=n, source=w,us report)*

SEGMENT END (5 CODERS)

3-11 In his speech on Monday, Mugabe thanked African leaders for refusing to buckle to pressure to suspend Zimbabwe from the Commonwealth of Britain and its former territories at a summit of the 54-nation grouping in Australia. *(on=thanked, only-factive=n, source=w,mugabe), (on=refusing to buckle, only-factive=n, source=w,mugabe,african leaders)*

Figure 2.5: An example document excerpt, with human **segmentations** and *annotations*.

which were produced separately from the clustering study, are also shown.

First, the segment consisting of sentences 3-8 through 3-10 seems to illustrate one potential way in which perspective coherence can be defined in terms of the sentence-level annotations: merge sentences into a segment when a single source (e.g., (w, us report)) is explicitly stating a sequence of opinions (e.g., (only-factive=n)). Note that segment boundaries thus occur where the previous and following sentences are discontinuous with respect to this type of coherence. In our example, the sources of the "ons" before and after the segment (that is, sentences 3-6 and 3-11) are different than the sources within the segment. As with other types of segmentation, linguistic phenomena such as the use of "his" in 3-11 to refer to Mugabe (who was most recently mentioned outside the segment) lends further support to such a segmentatation analysis. While this example shows one

way of abstracting over properties given our current sentence-level annotations, we believe that other abstractions will also be useful.

Note that sentence 3-7 provides an interesting borderline case, as one coder also included this sentence in the segment. First, there was no explicit "on". Second, sentences 3-8 through 3-10 can be seen as providing evidence for the expressive-subjective element "chaos." We hypothesize that the treatment of sentences whose content can be related by particular types of "informational relationships" (as discussed above) might impact perspective segmentation. For example, a more sophisticated notion of perspective coherence might be to cluster evidence together (as with sentences 3-8 to 3-10), then include it with the sentence(s) expressing the opinion that the evidence supports (sentence 3-7). Another possibility might be to ignore the presence of factive sentences that are providing evidence for an opinion, when trying to merge a sequence of opinionated sentences into a larger segment. An informal analysis of our data suggests that when evidence is treated the same way by coders, segment boundary agreement is about 60%.

We hope to further develop this notion of perspective coherence in future work, as we believe that segmentation will prove useful for higher level tasks. Just as in information retrieval, segmentation can be used to restrict question-answering and clustering to regions of documents, rather than whole documents. Our high level summaries might also want take advantage of segmentation, for example, by only including information from segments that either fully or partly convey subjective information. Finally, we believe that segmentation will help us generate new contextual features for our machine learning experiments; related notions such as density have already been shown to be useful in previous work.

2.6 Conceptualization of Perspective In Language

A conceptualization of perspective in language was prepared, which elaborates upon concepts used in the annotation instructions. In considers pragmatic influences on perspective, such as the identify and attitudes of a person, as well as the genre of the product and the context of the writing. It classifies various lexical clues of subjectivity along a number of dimensions. In addition, it considers how subjective expressions may be combined to form larger discourse structures. The conceptualization is a separate deliverable.

2.7 Repository of Linguistic Clues of Perspective

A number of promising linguistic clues of perspectives have been gathered together at MITRE in the directory /workshops/multip/lib. They come from existing published literature, as well as from automated processes.

2.7.1 Manually Identified Features

Five sets of manually-identified features were constructed during the MPQA Workshop, ranging from lists of likely *speech event* verbs, adjectives, and nouns; to lists of *psychological* verbs (*psych verbs*); to lists of discourse markers. With the exception of one feature set, which was manually entered from a published book, the sets were constructed from freely available electronic resources. The five basic sets are as follows:

1. **lev-speech** Speech event verbs compiled from an electronically available index of verbs and their classes, from (Levin, 1993).

2. **lev-fn-se** Verbs, nouns, and adjectives which are likely indicators of speech events from *The Framenet Project* (Framenet, 2002), combined with the verbs in 1.

3. **lev-fn-psych-plus** Psych verbs from (Levin, 1993) and psych verbs and adjectives from (Framenet, 2002). Some verb entries also contain the attribute *polar*, indicating whether the verb can be taken to reflect a positive or negative attitude.

4. **ballmer-se** This large list of possible speech event verbs was manually typed and is from (Ballmer and Brennenstuhl, 1981).

5. **disc-markers** A list of discourse markers obtained on-line from (Cues and for the Reader, 2002).

For each of these five sets, (at least) one file was created. Where more than one file exists, they are numbered (e.g. *disc-markers1*, *disc-markers2*, etc.), with the larger number indicating the latest version (typically with new attributes or errors removed). All the files have the same format of comments describing the feature set at the top of the file, followed by one entry per line, indicating various attributes of the feature, such as its type, length, spelling, part-of-speech, etc. For example, the first entry from *lev-speech* is for the word *ask*, and is as follows:

```
type=str_se_verb len=1 word1=ask pos1=verb stemmed1=yes bl_sec=37.1
```

For each of the five sets of features, a somewhat more detailed description is given below, taken from the documentation that accompanies the feature files.

- **lev-speech** Likely speech event verbs from Beth Levin's "English Verb Classes and Alternations" book. bl_sec indicates which section of book taken from, 37 is "verbs of communication," 33 is "judgement verbs," and 29 is (part of) "verbs with predicative complements," including appoint 29.1, characterize 29.2, dub 29.3, declare 29.4 , and conjecture 29.5 verbs. 37 taken as most likely speech event, therefore marked str_se_verb "strong speech event verb," section 33 is mod_se_verb=moderate, and 29 have the least likelihood (weak=wk_se_verb). Important notes: 1) these lists are

incomplete (i.e., not all English speech event verbs are covered) 2) words overlap between categories and sections.

- **lev-fn-se**

  This file contains likely speech event related verbs, adjectives, and nouns, from Beth Levin's "English Verb Classes and Alternations" book (verbs only) and from Framenet (verbs, nouns, and adjectives). Types beginning with bl are Levin's, fn are Framenet. There are 666 entries of which 619 are unique (i.e., no other entry has same word with same POS).

  Framenet entry notes: types are of form: fn_domain_frame_se_{v,a,n}. Words were obtained by searching for lemmas by frame element (*message*, *speaker*, and *topic* were used, so as to cover all of framenet's communication frames). Entries are not given likelihoods of being good speech event indicators, but the types should help, e.g., those with domain "communication" should be much better clues than those with domain "body" (in fact, these should probably be removed, but were left in for completeness).

  Levin entry notes: bl_sec indicates which section of book taken from, 37 is "verbs of communication," 33 is "judgement verbs," and 29 is (part of) "verbs with predicative complements," including appoint 29.1, characterize 29.2, dub 29.3, declare 29.4 , and conjecture 29.5 verbs. 37 taken as most likely speech event, therefore marked str_se_verb "strong speech event verb", section 33 is mod_se_verb=moderate, and 29 have the least likelihood (weak=wk_se_verb).

- **lev-fn-psych-plus**

  This file (lev-fn-psych-plus2.tff) contains psych verbs and some other verbs and adjectives, which are likely very good subjectivity indicators from Beth Levin's "English Verb Classes and Alternations" book (verbs only) and from Framenet (verbs and adjectives). Types beginning with bl are Levin's, fn are Framenet. There are 530 entries.

  Regarding duplicates: None of the entries are complete duplicates–they all differ at least in terms of one of the "type," "word1," or "pos1" attributes. Ignoring the type attribute, there are 419 entries which are unique (meaning for any one of these 419, there is no other entry with both the same word (field 3) and the same pos (field 4)). To isolate these 419 unique entries, simply type "sort -u -k 3,4" on the file.

  Framenet entry notes: types are of form: fn_domain_frame_se_v,a words were obtained by searching for lemma's with frame element: experiencer.

  Levin entry notes: bl_sec indicates which section of book taken from, as do the types. Types are bl_psych_verb, bl_judge_verb (judgement), and bl_desire_verb. Note that

bl_judge_verbs also overlap with those in other (speech event) feature files, namely lev-fn-se1 and lev-speech

Modification description: added new attribute "polar" with possible values "pos,neg,unk" (positive, negative, unknown) so that Levin verbs with such indications can be so notated. These are given only for sections 31.2 (Admire verbs) and 33 (Judgment verbs). There are 418 marked "unk" (i.e., from Framenet, or where Levin did not make an indication), 52 marked "pos," and 60 marked "neg."

- **ballmer-se**

This file (ballmer_se2.tff) contains "speech activity" verbs (over 6,000 of them) from pages 71–167 of the book: "Speech Act Classification: A Study in the Lexical Analysis of English Speech Activity Verbs" by Th. Ballmer and W. Brennenstuhl, 1981, Springer-Verlag.

The types given are the categories from the book, such as "type=EM2aa_Indicators_of_the_Emotional_Process" for the word "blush." In general, the first word in each entry was marked with pos=verb, and all others were marked as pos=unspec. Also, all stemming is yes, i.e., stemmed=y. The authors indicate for 302 of the verbs whether they were positive or negative, or sometimes other, non-polar values. So, the polar_plus attribute was created which contains these values. Verbs for which there was no indication are so indicated, with polar_plus=none.

Additional Notes: Bruce Fraser identified this book as a good source for speech events, and Dee DeLorenzo typed all of the verbs in so that an electronic version could be created. There are guaranteed to be duplicates in this file, if the attribute "type" is ignored (i.e., entries where the words are the same). Also note that the only non-alphanumeric characters present are {_ - '}.

- **disc-markers**

This file (disc_mrkrs1.tff) contains discourse markers taken from the web page: http://www.mapnp.org/library/writing/cuestran.htm which says: *Cues and Transitions for the Reader, Contributed by Deane Gradous, Twin Cities consultant Entered by Carter McNamara, PhD., Applies to nonprofits and for-profits unless noted.*

The types given are the categories from the web page, such as "to show addition" or "to contrast." Note that some of these would not likely be considered discourse markers, such as "imagine this scene" and the list is likely far from complete, although there are quite a few (172). Finally, note that some of the markers are actually patterns with gaps, such as "neither ... nor." These are marked with each gap counting as a single word called GAP_WD, e.g.: *word1=neither pos1=unspec stemmed1=y word2=GAP_WD pos2=unspec stemmed2=y word3=nor pos3=unspec stemmed3=y.*

2.7.2 Nouns Identified with Information Extraction Techniques

We section focuses on two types of features learned automatically: nouns and extraction patterns. First, we explain the motivation for learning these types of features. Next, we briefly overview the three learning algorithms that we used: AutoSlog-TS, Basilisk, and Meta-Bootstrapping. Finally, we explain how we ran the experiments and show some of the features that were learned.

Noun Features

Noun features are words that may be associated with subjectivity when they are used as a noun. Most of the subjectivity features studied in the past have been adjectives and verbs (Wiebe, 2000; Wiebe et al., 2002; Bruce and Wiebe, 1999; Hatzivassiloglou and Wiebe, 2000), but nouns can also be strong indicators of subjectivity. For example, many nouns are verb nominalizations, such as "complaint" or "discrimination". Many nouns also represent states associated with subjectivity (adjectival nominalizations, if you will), such as "happiness" and "ferocity".

Extraction Pattern Features

Many subjective expressions cannot be adequately captured by a fixed word sequence. We have identified several types of expressions for which N-gram representations are inadequate:

1. **Intervening syntactic constituents:** expressions that allow a noun phrase (usually a direct object) to be inserted.

Core expression:	*drove up the wall*	Examples:	*drove John up the wall*
General Pattern:	*drove [NP] up the wall*		*drove the mayor up the wall*
Core expression:	*talked to death*	Examples:	*talked John to death*
General Pattern:	*talked [NP] to death*		*talked the mayor to death*

2. **Intervening modifiers:** expressions that allow arbitrary modifiers to be inserted.

Core expression:	*step on toes*	Examples:	*step on John's toes*
General Pattern:	*step on [modifiers] toes*		*step on the mayor's toes*
Core expression:	*took interest in*	Examples:	*took a strong interest in*
General Pattern:	*took [modifiers] interest in*		*took a keen interest in*

3. **Intervening syntactic constituents and modifiers:** expressions that allow both a noun phrase and arbitrary modifiers to be inserted.

29

Core expression:	*gave a look*	Examples:	*gave Mary a dirty look*
General Pattern:	*gave [NP] a [modifiers] look*		*gave John a mean look*
Core expression:	*brought to knees*	Examples:	*brought Jim to his knees*
General Pattern:	*brought [NP] to [modifiers] knees*		*brought Sue to her knees*

In general, many expressions are flexible enough to allow for slight variations, including arbitrary direct objects and noun phrase modification. This flexibility cannot be adequately captured by N-grams because of their fixed nature. However, these variations can be modeled naturally using a syntactic representation. Extraction patterns provide exactly this type of syntactic flexibility. The extraction patterns that we used represent 13 types of syntactic patterns, which are shown in Table 2.4. These syntactic patterns are instantiated to represent specific expressions, such as *"complained about <np>"* or *"<subj> is a jerk"*. When used for information extraction purposes, the bracketed noun phrases are extracted as phrases of interest[4], but we used only the patterns and not the extractions themselves for our subjectivity experiments.

<subj> **passive-verb**	**active-verb** *<dobj>*	**noun prep** *<np>*
<subj> **active-verb**	**infinitive** *<dobj>*	**active-verb prep** *<np>*
<subj> **verb infinitive**	**verb infinitive** *<dobj>*	**passive-verb prep** *<np>*
<subj> **auxiliary noun**	**gerund** *<dobj>*	**infinitive prep** *<np>*
	noun auxiliary *<dobj>*	

Table 2.4: Syntactic representation used by the extraction patterns

In addition to syntactic flexibility, another benefit of extraction patterns over N-grams is that extraction patterns can distinguish between different verb voices, which is important because expressions can have different meanings in the active and passive voice. For example, you can say that a comedian "bombed" last night, which is a subjective statement, but you can't express this sentiment with the passive form of "bombed". In past work, we have found that distinguishing active and passive forms of the same verb can produce dramatically different results (Riloff, 1995; Riloff and Lehnert, 1994).

The Learning Algorithms

We experimented with three learning algorithms: (1) AutoSlog-TS, which generates extraction patterns, (2) Basilisk, which generates nouns, and (3) Meta-Bootstrapping, which generates both nouns and extraction patterns. In this section, we will briefly overview each algorithm.

---

[4] *subj* = the subject, *dobj* = the direct object, *np* = a noun phrase

30

AutoSlog-TS

AutoSlog-TS' patterns use a shallow parser called Sundance[5] to identify syntactic constituents (e.g., NPs and VPs) and to assign syntactic properties (e.g., active/passive voice and subject/object). Sundance can also apply AutoSlog-TS' patterns to extract information from text. The syntactic representations used by AutoSlog-TS' patterns were shown in Table 2.4.

AutoSlog-TS generates extraction patterns by using a *preclassified* text corpus, consisting of one set of text that is associated with the domain of interest (the "relevant text") and one set of text that is not associated with the domain of interest (the "irrelevant" text). The texts themselves do not need to be annotated in any way. For the subjectivity experiments, the relevant text was a collection of subjective sentences and the irrelevant text was a collection of objective sentences.

AutoSlog-TS ranks the list of extraction patterns based upon their strength of association with the relevant text (for details, see (Riloff, 1996)). In previous work, a human manually reviewed the top-ranking patterns to select the patterns that were truly reliable. For our subjectivity experiments, we did not manually review the patterns but collected all patterns that received a score higher than a threshold.

Basilisk

Basilisk is a bootstrapping algorithm that generates noun phrases associated with a semantic category (Thelen and Riloff, 2002). The input to Basilisk is an unannotated text corpus and a few manually defined *seed words* for the semantic category of interest. We used Basilisk to learn noun phrases associated with subjectivity, so we chose seed words that are highly subjective terms.

Before bootstrapping begins, AutoSlog-TS is applied to the corpus exhaustively so that an extraction pattern is produced to extract every noun phrase in the corpus. The bootstrapping process begins by selecting a subset of the extraction patterns that tend to extract the seed words. This is called the *pattern pool*. The nouns extracted by these patterns become candidates for the lexicon and are placed in a *candidate word pool*. Basilisk scores each candidate word by gathering all patterns that extract it and measuring how strongly they are are associated with the seed words. The five best candidate words are added to the lexicon, and the process starts over again. The output is a ranked list of nouns which are believed to belong the same semantic class as the seed words (in this case, subjectivity).

Meta-Bootstrapping

---

[5]Sundance was developed at the University of Utah.

The third feature generator that we used is the meta-bootstrapping algorithm (Riloff and Jones, 1999), which learns both noun phrases and extraction patterns associated with a semantic category. As its input, meta-bootstrapping requires an unannotated text corpus and a few "seed words" that represent the category of interest (subjectivity). For our experiments, we used highly subjective terms as seed words.

The heart of the meta-bootstrapping algorithm is a mutual bootstrapping process, which alternately learns an extraction pattern from the seed words and then uses the extraction pattern to learn new seed words. As an example, consider the word *martyrs*, which clearly has subjective overtones in almost any context and is therefore a good seed word. In the MUC-4 terrorism corpus, the word *martyrs* appears in 7 sentences that would yield the following extraction patterns: *"blood of <np>"*, *"weep for <np>"*, *"glory to <np>"*, *"look like <np>"*, *<np> have given lives"*, *"<np> deserve solidarity"*, and *"list of <np>"*. Several of these patterns are good indicators of subjectivity, but some are not. The bootstrapping process will select the pattern that has the strongest association with subjective noun phrases. For example, if the pattern *"glory to <np>"* extracts the words "martyrs", "heroes", and "liberators" then we might assume that it often extracts subjective phrases. All of its extractions would be added to the seed word list as subjective noun phrases, and the process repeats. The meta-bootstrapping algorithm also includes a second layer of bootstrapping that selectively chooses new seed words, which makes the algorithm more robust.

Results

Our first experiment used AutoSlog-TS to learn extraction patterns associated with subjectivity. To create a training corpus, we gathered sentences that workshop participants had manually labeled as clearly subjective or clearly objective. The corpus contained 936 subjective sentences and 410 objective sentences.

Table 2.5 shows the top 50 extraction patterns generated by AutoSlog-TS. Some of the patterns clearly represent subjective expressions, such as *"support for <np>"* and *"expressed <dobj>"*. But some patterns do not represent subjective expressions, such as *"one of <np>"* and *"sales to <np>"*. When analyzing the results, we concluded that the training set for AutoSlog-TS was too small to produce meaningful statistics. Although 6493 unique extraction patterns were generated, 6428 (99%) of them occurred with frequency ≤ 10. As Table 2.5 shows, we are encouraged that AutoSlog-TS did generate patterns representing a variety of subjective expressions. But we believe that a much larger training corpus will be needed to fully realize AutoSlog-TS' potential.

As input, the Basilisk and Meta-Bootstrapping algorithms require a corpus of unannotated texts and a handful of seed words for the category of interest. We used the same training corpus and seed words for both of these algorithms. The training corpus consisted of 2851 FBIS articles, which we gathered from the workshop's FBIS text collection by se-

<subj> said	*<subj> made*	*said <dobj>*
<subj> become	*become <dobj>*	*taiwan in <np>*
said in <np>	*one of <np>*	*<subj> think*
relations with <np>	*United_States in <np>*	*Taiwan is <dobj>*
independence by <np>	*support for <np>*	*<subj> ties*
sales to <np>	*weapons to <np>*	*<subj> like*
like <dobj>	*use of <np>*	*<subj> took*
threat to <np>	*Taiwan as <np>*	*<subj> declared*
to help <dobj>	*<subj> do*	*part of <np>*
war against <np>	*<subj> wants*	*ties with <np>*
expressed <dobj>	*<subj> believe*	*make <dobj>*
<subj> issue	*<subj> told*	*sale to <np>*
<subj> takes	*made in <np>*	*<subj> forces*
<dobj> told	*<subj> remains*	*one in <np>*
<subj> expressed	*<subj> defend*	*to defend <dobj>*
defend <dobj>	*<subj> clear*	*to avoid <dobj>*
sale of <np>	*China in <np>*	

Table 2.5: Top 60 extraction patterns generated by AutoSlog-TS

lecting all articles from June 2001. As seed words, we needed a small set of nouns that are strongly subjective and frequently occur in the FBIS texts.[6] To identify appropriate seed words, we began with the list of potentially subjective unigrams that Wiebe had compiled from her research. For each unigram (word), we counted the frequency with which the word appeared in the FBIS texts and sorted this list by frequency counts. As seed words, we chose the 20 words that appeared most frequently in our FBIS training corpus and that were used primarily as nouns. These 20 nouns are shown in Table 2.6.

lie	bet	win	fun	need
opposition	refuge	hope	critic	spite
harm	greed	danger	risk	pressure
friend	doubt	fear	fan	pain

Table 2.6: Subjective seed words used by Basilisk and Meta-Bootstrapping

Table 2.7 shows the top 90 nouns that were identified as subjective by Basilisk. The nouns toward the top of the list are strongly subjective (e.g., "scepticism", "derogation", "moral") and there are many subjective terms throughout the list. However, the quality of the terms tended to decrease as the bootstrapping progressed, which is a common trait of bootstrapping algorithms. One further avenue for investigation would be to put a human "in the loop" to review each proposed term as it is generated and give immediate feedback on whether the term should be considered subjective or not. This would keep the bootstrapping process on track and hopefully lead to the generation of more high-quality terms. Overall, Basilisk's results look promising and we believe that additional experiments with Basilisk are worthy of further investigation.

The third algorithm that we tried was meta-bootstrapping, which learns both nouns and extraction patterns. Table 2.8 shows the top 100 nouns and Table 2.9 shows the top 60 extraction patterns that were produced by meta-bootstrapping. Meta-bootstrapping performed by far the best of the three feature generators that we tried. As Table 2.8 shows, nearly all of the top nouns seem to be highly subjective. Many subjective nouns were generated farther down on the list as well. Meta-bootstrapping generated a large number of noun features that seem to be extremely good indicators of subjectivity. Many of the extraction patterns generated by meta-bootstrapping also seem to represent highly subjective expressions, such as *"disagreement with <np>"* and *"stressed <dobj>"*. Many of the extraction patterns represent verb phrases associated with statements and opinions.

In summary, we are encouraged by these experiments which suggest that many subjective noun features and extraction pattern features can be generated using automatic

---

[6]If the seed words do not appear very often in the training set, then the bootstrapping process will have difficulty gaining any momentum.

scepticism	derogation	skepticism	eight	stigmatisation
chunks	moral	sevenfold	p'yong	anxieties
golkar-affiliated	bits	tumors	indignation	credence
induction	chub	rundown	bewilderment	ruling-off
responsiveness	certainty	qureshi	chin	vectras
senqu	transitions	mi-8	such	creuzfeldt-jakob
creutzfeldt-jakob	creutzfeldt-jacob	barrage	butchers	equipping
40's	abepura	jeopardy	kaliningrad	icu
classification[8	albairate	aerospace	ten-thousand	hundred-thousand
precariousness	savoir-faire	mess	huvsgul	bordeaux
khomasdal	haste	arkhangelsk	formative	invasions
malabo	tagtabazar-2	pocket	coimbra	merry-go-round
prudence	intrigue	kickbacks	overbiddings	messianism
indication	saga	genome	teeth	farther
reponsibility	arrondisement	sunnah	hooliganism	bas-congo
turnabout	practicalities	vary	dysfunction	zia
kasai	feedback	vices	jealousy	recurrence
kazakhgates	leaks	recurring	importations	validation

Table 2.7: Top 90 nouns generated by Basilisk

35

puppets	affect	concern	obstacles	challenges
conspiracies	threats	threat	opinion	balance
problem	piet	question	chief	military
prediction	views	horror	opinions	forms
scepticism	skepticism	derogation	indignation	credo
anxieties	bewilderment	astonishment	ignorance	regret
discontent	alarm	dissatisfaction	condolences	gratitude
abidance	doubts	happiness	satisfaction	admiration
appreciation	conviction	anxiety	adherence	pledge
shock	malherbe	fears	feeling	allegiance
reservations	sympathy	commitment	ambitions	sentiments
intentions	wish	desire	willingness	anger
solidarity	pride	depth	belief	concerns
steadfastness	surprise	sentiment	wishes	intention
readiness	relief	capabilities	sincere	variety
determination	disagreement	importance	confidence	victory
will	necessity	understanding	initiatives	interest
supremacy	tolerance	merger	call	remarks
stakes	support	idea	demand	approval
presence	spirit	cause	intent	initiative

Table 2.8: Top 100 nouns generated by Meta-Bootstrapping

confront <dobj>	reiterated <dobj>	disagreement with <np>
to convey <dobj>	please <dobj>	Turkmenistan has <dobj>
poses <dobj>	voiced <dobj>	agency to <np>
to pose <dobj>	stressed <dobj>	<subj> was expressed
<subj> was resolute	affirmed <dobj>	was created on <np>
noting <dobj>	stand by <np>	peace on <np>
was impressed by <np>	minimize <dobj>	expressed <dobj>
well-being of <np>	presenting to <np>	stress <dobj>
tasks of <np>	reaffirmed <dobj>	trying to make <dobj>
sticking to <np>	tribute for <np>	to show <dobj>
dimensions of <np>	pose <dobj>	<subj> make effort
relations are <dobj>	nothing about <np>	expressing <dobj>
<subj> chaired session	wish of <np>	answered <dobj>
presenting to <np>	underlined <dobj>	posed <dobj>
<subj> deserves	underscored <dobj>	reconsider <dobj>
supportive of >np>	reaffirms <dobj>	express <dobj>
sticking to <np>	<subj> requesting	directions to <np>
expresses <dobj>	stressing <dobj>	<subj> deserves attention
formed on <np>	think on <np>	participating for <np>
play <dobj>	raises <dobj>	<subj> playing a game

Table 2.9: Top 60 extraction patterns generated by Meta-Bootstrapping

learning algorithms. We could not draw any conclusions about AutoSlog-TS as a feature generator because the training set that we used was too small, but both Basilisk and Meta-bootstrapping produced intriguing results. Meta-bootstrapping performed the best, generating a large number of subjective noun features as well as subjective extraction pattern features, and Basilisk generated some strongly subjective noun features as well. These results warrant additional experiments to see if these algorithms can be employed to even greater advantage, for example by using bigger and perhaps better seed words lists and training corpora.

2.7.3 Adjectives and Verbs Automatically Learned from Text

This section describes work using the process reported in (Wiebe, 2000) for learning subjective adjectives and verbs to learn larger sets of features than reported in the original paper. The current work applies the process to much more data, which is annotated only at the document level. Specifically, eight files from the Treebank corpus were used, for a total of 1,270,665 words (W9-2, W9-20, W9-21, W9-23, W9-4, W9-10, W9-22, and W9-33). The document-level classes are specified by the Wall Street Journal itself. That is, we define the class *opinion-piece* to be the union of *Editorials, Letters to the Editor*, *Arts & Leisure*, and *Viewpoints*.

The criterion for selecting a set S of adjectives as good clues of subjectivity (i.e., of opinions) is the the precision of S with respect to opinion pieces. This is defined as:

$$prec(S) = \frac{\text{number of instances of members of S in opinion pieces}}{\text{total number of instances of members of S in the data}}$$

This metric is used during all three phases of training, validation, and testing.

The list of adjectives produced for the workshop is the union, over the eight datasets listed above, of the adjectives produced to test on each dataset. For each test set, multiple training-validation dataset pairs were used.

No manual editing of the list has been performed.

The approach is based on *distributional similarity*, where words are judged to be more or less similar based on their distributional patterning in text.

Distributional similarity is most commonly used in NLP for two purposes: to create dictionaries and thesauri from corpora (see, for example, (Lin, 1998; Riloff and Jones, 1999)) and to smooth parameter estimates of rare or unseen events to improve syntactic or semantic disambiguation (see, for example, (Hindle, 1990; Dagan, Pereira, and Lee, 1994)). The procedure presented below for learning PSEs with distributional similarity involves both.

Many variants of distributional similarity have been used in NLP (see (Lee, 1999; Lee and Pereira, 1999) for comparisons of a number of methods). Dekang Lin's (1998) method is used in this work. In contrast to many implementations, which focus exclusively

on verb-noun relationships, Lin's method incorporates a variety of syntactic relations. This is important for subjectivity recognition, because PSEs are not limited to verb-noun relationships. In addition, Lin's results are freely available.

Using his broad-coverage parser (Lin, 1994), Lin (1998) extracts dependency triples from text which consist of two words and the grammatical relationship between them: $(w1, relation, w2)$. To measure similarity between two words $w1$ and $w2$, $T(w1)$ and $T(w2)$ are identified, where $T(w)$ is the set of relation-word pairs correlated with w. The similarity $sim(w1, w2)$ between two words $w1$ and $w2$ is then defined as (where $I(x, r, y)$ is equal to the mutual information between words x and y):

$$\frac{\sum_{(r,w)\in T(w1)\cap T(w2)}(I(w1,r,w)+I(w2,r,w))}{\sum_{(r,w)\in T(w1)} I(w1,r,w) + \sum_{(r,w)\in T(w2)} I(w2,r,w)}$$

Lin processed a 64-million word corpus of news articles, creating a thesaurus entry for each word consisting of the 200 words of the same part of speech that are most similar to it.

As mentioned above, distributional similarity is typically used for one of two purposes: (1) creating dictionaries and (2) smoothing parameter estimates.

Consider (1), which is Lin's focus. The intuition behind his method is that words correlated with many of the same words are more similar. We hypothesized in this work that these words might be distributionally similar because they share pragmatic usages, such as expressing subjectivity, even if they are not close synonyms. For example, consider the 20 most similar words to the adjective **bizarre**: *strange, similar, scary, unusual, fascinating, interesting, curious, tragic, different, contradictory, peculiar, silly, sad, absurd, poignant, crazy, funny, comic, compelling, odd.* Some of these are relatively close synonyms, e.g., *strange, unusual, curious, peculiar, absurd, crazy, odd.* Others, while not close synonyms, are also subjective, e.g., *tragic, sad, poignant, compelling.* We would like to identify those as well. Thus, we attempt to extend the set of candidate PSEs beyond those in the training data, by considering words similar to those in the training data.

Now consider (2), smoothing parameter estimates. Evidence is given in the larger article of which this is a part that low-frequency and unique words appear more often in subjective texts than expected. Thus, we do not want to discard low-frequency words from consideration, but cannot effectively judge the suitability of individual words. To decide whether to retain a word as a PSE, we consider the precision not of the individual word, but of the word together with its cluster of similar words. A set of seed words begins the process. For each seed s_i, the precision of the set $\{s_i\} \cup C_{i,n}$ in the training data is calculated, where $C_{i,n}$ is the set of the n words that are most similar to s_i. If the precision of $\{s_i\} \cup C_{i,n}$ is greater than a threshold T, then the words in this set are retained as PSEs. If it is not, neither s_i nor the words in $C_{i,n}$ are retained. The union of the retained sets

$trainingPrec(s)$ is the precision of s in the training data
$validationPrec(s)$ is the precision of s in the validation data
$testPrec(s)$ is the precision of s in the test data
(similarly for $trainingFreq, validationFreq$, and $testFreq$)
S = the set of all adjectives in the training data
for T in $[0.01, 0.04, \ldots, 0.70]$:
    for n in $[2, 3, \ldots, 40]$:
       $retained = \{\}$
       For s_i in S:
          if $trainingPrec(\{s_i\} \cup C_{i,n}) > T$:
             $retained = retained \cup \{s_i\} \cup C_{i,n}$
       $R_{T,n} = retained$
$ADJ_{pses} = \{\}$
for T in $[0.01, 0.04, \ldots, 0.70]$:
    for n in $[2, 3, \ldots, 40]$:
       if $validationPrec(R_{T,n}) \geq 0.28$ (0.23 for verbs)
       and $validationFreq(R_{T,n}) \geq 100$:
          $ADJ_{pses} = ADJ_{pses} \cup R_{T,n}$

Figure 2.6: Algorithm for selecting adjective and verb features using distributional similarity

will be notated $R_{T,n}$, that is, the union of all sets $\{s_i\} \cup C_{i,n}$ with precision on the training set $> T$.

In (Wiebe, 2000), the seeds (the s_i's) were extracted from the subjective-element annotations in a corpus. Specifically, the seeds were the adjectives that appear at least once in a subjective element in that corpus. In 10-fold cross-validation experiments, where only 1/10 of the data is used for training, and 9/10 is used for testing, we achieved an average increase of more than 13 percentage points over the baseline precision of the entire set of words in the test data. A small amount of training data was used to explore the idea that the process is appropriate even when little training data is available.

In this work, the opinion-piece corpus is used to move beyond the manual annotations and small corpus of the earlier work. The process is performed separately for adjectives and verbs (other parts of speech will be tested in future work). In addition, a much looser criterion is used to choose the initial seeds: all of the adjectives (verbs) in the training data are used.

The process for adjectives is given in algorithmic form in Figure 2.6. (The process is

the same for verbs, with one small difference noted in the figure.) Seeds and their clusters are assessed on a training set for many parameter settings (cluster size n from 2 through 40, and precision threshold T from 0.01 through 0.70 by 3). As mentioned above, each n, T parameter pair yields a set of adjectives $R_{T,n}$, that is, the union of all sets $\{s_i\} \cup C_{i,n}$ with precision on the training set $> T$. A subset, ADJ_{pses}, of those sets is chosen based on precision and frequency in a validation set. Finally, the ADJ_{pses} are tested on the test set.

The higher precision of the set of identified adjectives for four test datasets is given in the journal article of which this is a part. To test the adjectives generated for the test sets, multiple training-validation dataset pairs (that are distinct from the test dataset) were used for each test set. For a given test set, the union is formed of the adjectives identified from each training-validation pair. The list of adjectives available on this page is the union of all those sets, over the eight test datasets.

2.7.4 Collocations Learned from Text

Collocational features learned during previous work by workshop participants were also made available to the workshop. This work was originally reported in (Wiebe, Wilson, and Bell, 2001).

Collocations were mined from the three corpora based on the following method. First, all 1-grams, 2-grams, 3-grams, and 4-grams were extracted from the training data, and the precision of each was calculated. The precision of an n-gram is the number of subjective instances of that n-gram divided by the total number of instances of that n-gram. An instance of an n-gram is subjective if each word occurs in a subjective element. As mentioned above, boundaries between subjective elements are ignored, so there is no restriction that all words of the n-gram appear in a single subjective element.

Potentially subjective collocations were selected based on their precision, using two criteria. First, the precision of the n-gram must be at least 0.1. Second, the precision of the n-gram must be at least as good as the precisions of its constituents.

For example, let $(W1, W2)$ be a bi-gram consisting of consecutive words $W1$ and $W2$. $(W1, W2)$ is identified to be a potential subjective element if $precision(W1, W2) > .1$ and (where pc is $precision$):

$$pc(W1, W2) > max(pc(W1), pc(W2))$$

For tri-grams, we extend the second condition in the following way. Let $(W1, W2, W3)$ be a tri-gram consisting of consecutive words $W1$, $W2$, and $W3$. Then the condition is:

$$pc(W1, W2, W3) > max(pc(W1, W2), pc(W3)) \text{ or}$$

$$pc(W1, W2, W3) > max(pc(W1), pc(W2, W3))$$

4-grams were selected in the same manner as 3-grams, comparing the 4-gram with first the maximum of the precisions of word $W1$ and tri-gram ($W2$, $W3$, $W4$) and then with the maximum of the precisions of trigram ($W1$,$W2$,$W3$) and $W4$. The n-gram collocations identified as above will be called *fixed-n-grams*, i.e., they are fixed sequences of words of length n.

These fixed collocational features show moderate improvements in precision for subjectivity tagging.

In future work, we intend to apply other features learned in previous work to the data and tasks of the MPQA project. In particular, very low frequency words (words unique in, say, a 650K word corpus) are more likely to be in subjective tasks then one would expect. In addition, combining the process above for learning fixed combinations with a *unique* constraint results in the highest-precision features we have discovered to date. A modest amount of work is required to adapt the learning process so these types of features can be used in the MPQA project.

2.8 Learning Architecture

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

- to facilitate the use of MPQA 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.

This section describes the learning architecture. We begin with an overview of its structure, and elaborate on the overview in the subsections below.

In order to accomodate data from annotated training documents and a variety of feature generators, the architecture is organized around a general database for storing information about documents. The database stores both the document text and additional information extracted from, added to, or about the document. This additional information is stored as *annotations*, which are records that are logically attached to a portion of the text. The document/annotation database is detailed in section 2.8.1.

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", 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 system of generator programs, coupled with the annotation representation, and the database, provides a flexible architecture for composing training data for learning. Feature generation is discussed in section 2.8.2, and instance generation is discussed in section 2.8.3.

Instance and feature annotations can be compiled together and converted to a form suitable for use as training data. In a series of preliminary experiments, 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 section 2.9. 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."

We conclude this overview with a discussion of some high-level design decisions and the motivation behind them.

- 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 architecture 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.)

2.8.1 Annotation Database

The goal of the database design is to store documents and annotations in a format that is both easily read (if necessary) by humans and easily implemented in programs and scripts. In the subsections below, we describe the organization of the database, and the format of the data files.

Database Organization

The database is partitioned into areas for storing different kinds of information. For each document in the database, there is a subdirectory within each area of the database. The parallel subdirectories taken together comprise all the information about the document. The following is a list of the areas, their contents, and their data formats:

database/docs/*path*/*document*

> Document text is stored in plain text files in the docs area. The *path* allows further structure to be imposed on the database. For the duration of the workshop, the database was structured into manually retrieved FBIS documents (temp_fbis); manually retrieved other documents (non_fbis); and paths for FBIS documents organized by date (e.g., 20010613). Each document has a unique name *document*, with parallel directories named *document* in the other areas of the database.

> As indicated in the overview, annotations are logically attached to portions of the text. These attachments are represented as spans (start and end indices) into the document text file. Like annotation files, the document text is immutable, so that spans may not become invalid.

database/gate_anns/*path*/*document*.xml

> This file is a GATE XML version of *document* for use in the annotation tool. Annotators load this XML document into GATE and annotate it; the annotations are subsequently transferred to the man_anns area and converted to MPQA format.

database/meta_anns/*path*/*document*/...

> Information encoded in the original source document is stored in the meta_anns area when it is extracted from the document to produce the raw text. This information might include, for example, the source, date, and title of the document. It is stored in MPQA format.

database/man_anns/*path*/*document*/...

> Manual annotations for a document are stored in the man_anns area in MPQA format. Each *document* directory may contain annotation files from multiple annotators, or even multiple files from a single annotator. File links with fixed names indicate the "official" version of the manual annotations, so that programs can locate it among the possible versions.

database/auto_anns/*path*/*document*/...

> Automatically generated annotations—including feature annotations—are stored in the auto_anns area in MPQA format.

Gate XML Format

GATE XML is the primary format of the GATE tool that we employed for annotation and document processing. This format encodes both the document text and annotations using XML structure. The text is marked with "Node" anchors to which the annotations refer. A sample GATE XML document is given below.

```
<GateDocument>
...
<TextWithNodes>
...
<Node id="234"/>a<Node id="235"/> <Node
id="236"/>military<Node id="244"/> <Node id="245"/> <Node
id="246"/>spokesman<Node id="255"/> <Node
id="256"/>said<Node id="260"/>.
...
</TextWithNodes>
<AnnotationSet>
  <Annotation Type="agent" StartNode="234" EndNode="255">
  <Feature>
    <Name className="java.lang.String">id</Name>
    <Value className="java.lang.String">spokesman</Value>
  </Feature>
  <Feature>
    <Name className="java.lang.String">nested-source</Name>
    <Value className="java.lang.String">w, spokesman</Value>
  </Feature>
  </Annotation>
  <Annotation Type="on" StartNode="256" EndNode="260">
  <Feature>
    <Name className="java.lang.String">nested-source</Name>
```

```
    <Value className="java.lang.String">w, spokesman</Value>
  </Feature>
  <Feature>
    <Name className="java.lang.String">onlyfactive</Name>
    <Value className="java.lang.String">yes</Value>
  </Feature>
  </Annotation>
</AnnotationSet>
</GateDocument>
```

MPQA Annotation Format

The MPQA annotation format is designed to be easy to read for both humans and programs. Annotations are represented each on a single line of a text file. Four initial fields give an annotation's internal ID, its span into the text, its type, and its name. Below is an example MPQA file.

```
100 234,255 string agent id="s" nested-source="w,s"
101 256,260 string on    nested-source="w,s" onlyfactive="yes"
...
```

The first annotation has ID 100 and spans from position 234 to 255 in the text. Its name is "agent." Internal IDs are unique only within the annotation file. Most annotations are of type "string," which means that the rest of the entry may be an arbitrary string ending with a newline. Many string annotations represent annotation attributes in an XML-style attribute format, as exemplified above.

The **gate2mpqa** utility extracts annotations from GATE XML documents and converts them to MPQA format.

2.8.2 Feature Generation

Feature generation programs typically take a document as input (both its text and its annotations if desired) and produce a new annotation file. The **mkauto_anns** utility runs a feature generator on each document in the database to populate the database with feature annotations.

Text Processing

The current implementation of the learning architecture includes a number of text processing components. Each of these components is tooled to produce MPQA annotations for the auto_anns area.

GATE Tokenization, Sentence Splitting, Part-of-Speech Tagging
These preprocessing components are executed together within GATE. The resulting GATE XML is converted to MPQA.

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 Porter's 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 additional 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. For example, the following pattern—

> 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

—matches "<u>What a bunch of</u> baloney!"

The **match_tff** utility applies a TFF specification to a document, creating an annotation file for the feature annotations matching the TFF specification.

The following is a current list of TFF feature specifications:

- Speech event verbs from Ballmer and Brennenstuhl (Ballmer and Brennenstuhl, 1981), from Levin (Levin, 1993), and from Framenet (Framenet, 2002).

- Psych verbs from Levin (Levin, 1993) and from Framenet (Framenet, 2002).

- Potential subjective element words and phrases from (Wiebe et al., 2002).

- Subjective patterns induced via the meta-bootstrapping process (Thelen and Riloff, 2002).

Cascades and Feedback

Annotations produced by learned classifiers may also be a source of features for subsequent learning. The learning architecture allows classifiers to add annotations to the database, providing the opportunity for training cascades of classifiers, or applying strategies involving feedback, such as active learning or co-training. Although we have not yet explored this possibility, we plan to do so in the future.

2.8.3 Instance Generation

As described in the overview, instance generator programs produce training and testing instances from the document database. This involves the following steps: first, an annotation file of potential instances is selected from the database; next, non-instances are filtered out from its annotations; finally, feature values are associated with each instance.

The current architecture includes a utility, **mpqa2arff**, that runs instance generators over the database. The resulting instances are represented in a standard format for machine learning called ARFF. The ARFF header indicates the attributes associated with each instance. Each subsequent line of the file gives the attribute values for a single instance.

2.9 Automatic Annotation

We performed several initial experiments to reproduce some of the manual annotations automatically. We chose to start with ons[7].

Programs were also written and executed to recognize expressive subjective elements and to automatically perform factivity judgements. This code is a good starting point for future work. The results are not presented here because the feature-selection method is primitive, and essentially all features are included. The current results are almost 100% recall, but low precision. Essentially, so many features are included without discrimination that the system thinks that all sentences are opinionated. In continuing and future work, we are refining these experiments.

The results for the "ons", however, are already promising.

Any machine-learning experiment needs a baseline, and for ours, we chose something very simple. If a word's lemma was found on one of two wordlists, we consider it to be

---

[7]The precise task is recognition of single-word, explicit 'on's (excluding the writer and any other implicit ons).

an on; other words and word-sequences are left unmarked. The two wordlists come from (Levin, 1993)[8] and from Framenet(Framenet, 2002)[9]. In both cases, these lists were chosen because they are the group in which *say* occurs, since *say* is essentially always an on.

For a machine-learning approach, we need four things. A set of features, annotated data, an algorithm, and an implementation. We used the Weka machine learning package, choosing its naive Bayes and k-nearest-neighbor algorithms[10]. We used all the data annotated at the time we ran the experiment.

The features we used were all words within 2 words on either side of the target word, the part of speech of the target word, the category from the two sources above (Beth Levin and Framenet). We also used some features derived from the CASS (Abney, 1996) partial parser – the category of the current word's chunk, of the previous chunk, and of the next chunk.

2.9.1 Metrics

Since recognizing ons is an extent-tagging task, we use the precision, recall, and F-measure metrics common in such tasks. Given the sets of entities G and S annotated in the gold-standard and by the system, respectively, we have Recall $= \frac{|G \cap S|}{|G|}$, Precision $= \frac{|G \cap S|}{|S|}$, and $F = \frac{1}{\frac{\frac{1}{P}+\frac{1}{R}}{2}} = \frac{2PR}{P+R}$.

2.9.2 Results

algorithm	Precision	Recall	F-measure
baseline	69.9	47.7	56.7
Naïve Bayes	46.7	76.6	58.0
kNN	69.6	63.4	66.4

Table 2.10: results, initial on learning experiments

Table 2.10 presents the results of the initial learning experiments. The machine-learning numbers use a 10-fold cross-validation (the baseline does not, because it doesn't involve training). We are pleased that by the F-measure statistic, both learning algorithms bested the baseline.

---

[8]Section 37.7

[9]From the Communication domain, the frame being statement speech event verbs.

[10]We explored other algorithms as well, but time pressures as well as the intent of the Weka package as a teaching tool kept us to our initial, fast-running choices.

2.10   End-User Evaluation

The final component of the MPQA workshop is the End-User Evaluation. There are three main goals of the End-User Evaluation. First, we want to explore what aspects of opinions are likely to be the most useful for accomplishing opinion tasks that would be of direct interest to analyst users. Next, we want to establish a framework for evaluating opinion tasks. Finally, we want to conduct an example evaluation to explore what obstacles will be faced in a full evaluation.

During the pre-workshop meeting and the workshop itself, a large number of opinion-related questions that analysts might be interested in were discussed. A partial list includes:

- Is an opinion being expressed?

- Who is expressing it?

- On whose behalf is the opinion being expressed?

- What is the type of opinion (e.g. religious or political)?

- Is the viewpoint dependent on the audience?

- Is the opinion consistent with past opinions of this agent?

- Is the tone consistent with past opinions of this agent?

- Are the opinions of the agent consistent with the opinions of any other particular group?

- Is this opinion different from the opinions of other larger groups that the agent belongs to?

- Given the past opinions of an agent, what will the opinion be for a projected event?

Most of the interesting questions regarding opinions involve grouping opinions of either a person or a larger organization and then detecting whether a given opinion (or opinions) is consistent with these grouped opinions. This suggests that the operation of accurately clustering opinions together will be crucial and should be the focus of initial investigations into end-user tasks.

The language, and therefore annotation scheme, needed to fully describe a given instance of opinion or subjectivity is very rich, as we have seen previously in this report. Almost all of this richness will eventually be needed by text analysis programs in order to represent an opinion instance for an analyst. However, it is clear that many opinion-related questions can be at least initially answered using a much less detailed analysis of

the opinions. This is the motivation for the development of the shallow annotation scheme presented earlier, where only indications of opinions, the agent expressing the opinion, and additional subjective language is identified. These shallow annotations will be the input and evaluation mechanisms for our end-user task.

2.10.1   How Do Humans Cluster?

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 workshop participants plus an ex-analyst (Penny Lehtola) manually clustered opinions from documents related to 3 topics:

1. Election in Zimbabwe

2. Treatment of prisoners at Camp X-Ray and Guantanamo Bay.

3. Bush 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:

> The criteria for clustering is entirely up to you, except it should be related to perspective and some analyst need. We'd like the output to be either one level of clusters, or two level of clusters (i.e., cluster the opinions found in the larger top-level cluster, possibly using a different criteria.)

We told participants to expect to spend 4-5 hours on the task, and to make sure they at least clustered the first topic ("Zimbabwe").

We held a video conference using MITRE's facilities to discuss the results. As would be expected given the lack of instructions, the participant background strongly influenced the type of clusters. The linguist separately clustered every sentence according to the perceived purpose of 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 a 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 commonly 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 lessons 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 clusters 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.

2.10.2   End-User Scenario

The overall user scenario in which our end-user evaluation task fits is a series of interactions between the user and the system.

Stage 1: User states a topic of interest and interacts with the IR system, possibly in multiple stages including relevance feedback, to identify a set of potentially relevant documents.

Stage 2: User states particular perspective question on topic. This question should identify source type (e.g., governments, individuals, writers) of interest and be a Yes/No (or Pro/Con) question for now. The system then clusters segments of documents that respond to the question based on the question and the document text along with the

documents' automatically derived perspective annotations. The goal is to group together document segments with the same answer and perspective (including expressive content).

Stage 3: User states constraints on clustered documents or segments. These might be geographic, temporal, ideological, political, or religious, for example. The system then shows sub-clusters or highlighted document segments so the user can get an impression (visual or statistical) of whether the constraints match clusters. For example, the user can determine whether certain geo-political groups share opinions or whether an agent's opinion has changed over time.

2.10.3   Document and Topic Collections

As expected, identification and construction of appropriate document collections for our particular task was a large undertaking. It is important for our task to have opinions about a given topic from multiple sources and expressed in different tones and word choices. Existing single source collections of documents and queries, for example, TREC sub-collections or REUTERS, are not useful since in general there are insufficient different opinions on a subject, and there is a uniform language style for all articles.

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 system. (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.

The FBIS collection is all English language, with 60% being translated by FBIS from a foreign source, with 84 different original languages. The remaining 40% were originally published in English, though in some cases the quoted opinions were originally spoken in some other language. 20% of the documents come from TV or radio. 5% are explicitly identified as editorials, though there are other editorials not identified.

The FBIS documents are freely available to any group with an existing government contract. Copyright issues prohibit making it more generally available. Note that the oft-repeated rumor that no foreign nationals are allowed to use FBIS is false. The only requirement that FBIS puts on the data is existence of the government contract. We are currently attempting to work out an arrangement to make our particular collection easily available to any group with such a contract.

The World News Connection (WNC) is a copyright clearinghouse that makes a subset of FBIS completely publicly available (for a small fee). It can be seen on the web at http://wnc.fedworld.gov. Obviously, documents available through WNC are more useful for research purposes since anybody can use them.

We have identified a small subset of 575 documents from our FBIS collection that is available through WNC. Our annotations and other small experiments are being done on

this subset of publicly available documents. We ran 8 topics on our full FBIS collection retrieving 200 documents each. We then identified 575 of those 1600 documents as being publicly available from WNC.

We are currently in negotiations with WNC to arrange to have WNC distribute both data and some of our annotations of the 575 document collection. This will require a one-time setup fee and then modest individual fees from each research group. The time-frame of public distribution is still being worked out.

For work within our workshop, we've constructed a set of 8 topic sentences, each consisting of a couple of clauses and meant to correspond to both the topic and question of Stage 1 and 2 of the end-user scenario. All 8 topics/questions are Pro/Con questions, for example, *"Was the 2002 election in Zimbabwe fair?"*. As stated above, we then ran these topics using SMART with relevance feedback on the full FBIS collection. We then identified 40-105 related documents per topic (not all relevant to the original topic) to define the WNC corpus.

For 4 of the 8 topics we have manually gone through all the related documents and identified segments that answered the Pro/Con topic. 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, what the answer is, and the start and ending location of the segment. Note that preparing these answer segments required little training to do, and took about 5 minutes per document.

2.10.4   Sample Simple Evaluation Task

The first end-user evaluation task is an initial implementation of stage 2 of the scenario described above. The goal is to evaluate whether our simple automatic identification of opinions is sufficient to improve clustering of opinions.

For each of four topics in the WNC collection, we find the single best passage within each related document that answers the topic. 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 trial, the determination of best passage and the clustering between passages is dependent on the terms within the candidate passages only. In the second trial, 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 previously. 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.

Two key implementation aspects of this process are what sort of candidate passages are

considered, and how the best passages are clustered. We implemented two algorithms for determining candidate passages. One was a simple static algorithm that targeted passages of length about 800 characters, broken on sentence boundaries. Overlapping passages were 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. The second algorithm we implemented attempted to find sets of related sequential sentences, including features such as shared terms and whether the sets cross paragraph boundaries. Unfortunately, this did not behave well, in part because of peculiarities of the document collection. Depending on the original source of each document, there might be paragraph boundaries at every sentence, or no paragraph boundaries except to separate the title from the rest of the article. So only the simple static algorithm was used for the results presented here.

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.

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.

In this experiment, clusters were merged in phase 2 until there were only 3 result clusters. There was an additional criteria that no cluster can contain more than 2/3 of the passages. This ensured that the result was not one huge cluster with 2 outlier passages forming their own clusters.

Table 2.11 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. Different passages were often chosen, but the passages sometimes included opinion indicators that were unrelated to the topic. This lack of coherency is a weakness of using static passages; this needs to be explored in future experiments.

	Base				Opinions			
Cluster	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

Table 2.11: Cluster Evaluation: Zimbabwe

The results of the Kyoto topic are given in Table 2.12. 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.

	Base				Opinions			
Cluster	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 2.12: Cluster Evaluation: Kyoto

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.

2.10.5   Simple 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 given by the topic answers themselves (we have perfect knowledge about relevant opinions.) Cluster these passages using the same algorithms as previously.

Tables 2.13 and 2.14 give the results for the same Zimbabwe and Kyoto topics 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 rather than use some other criteria.

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

Cluster	Both	Yes	No
1	1	1	18
2	0	1	10
3	0	23	34

Table 2.13: Retrospective Cluster Evaluation: Zimbabwe

Cluster	Both	Yes	No
1	1	6	19
2	0	2	1
3	0	4	7

Table 2.14: Retrospective Cluster Evaluation: Kyoto

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.

2.10.6   End-User Summary

We have demonstrated an end-to-end system that

- Retrieves documents from a large database

- Adds opinion annotations to these documents using automatic NLP tools

- Clusters passages partly based on those features

- Evaluates whether the clusters were successful.

Our algorithms for taking advantage of opinion annotations in the end-user task were not shown to be effective. In fact, most of the algorithms failed miserably. That is not terribly surprising given the timing constraints: the 575 documents in the WNC collection were identified during week 7, and the answer segments for them were constructed on Thursday of week 8 (the final week). Even if the algorithms had worked well, nothing could be concluded from the results since the experiment size was much too small.

However, the goal of the end-user task was not to present final solutions to the end-user needs but to establish an evaluation methodology in which solutions to those needs can be analyzed and evaluated. This was accomplished; the evaluation schemes were able to detect failure and success of our solution algorithms. Further work on the evaluation methodology is needed; the task of clustering opinions into 2 clusters is artificial and not sufficiently related to end-user needs. Our results show that we can evaluate clustering of opinions.

2.10.7   A Future End-User Task

In order to be able to draw scientific conclusions from work in this area, the task should be modified a bit and needs to be expanded significantly. One possible TREC-like experimental evaluation task that might be done in the future is described here.

Our FBIS collection (270,000+ documents) has shown itself to be a valuable source of varied opinions on many topics; it seems to be the best set of documents out there. The restrictions to government contractors is a major problem, though. Getting a small subset of FBIS through WNC as was done for our workshop seems to be a feasible solution to this problem.

The evaluation task organizers would have access to the FBIS collection and would construct 25-50 opinion topics that have substantial numbers of related documents (100+ per topic) within the FBIS collection. The organizers would then find a subset of these documents (30+ per topic) that are available from WNC. A CD containing these several thousand documents could be constructed by WNC and made available to participants of the evaluation task. The organizers would also construct answer sets, giving answer segments for each of these documents. Note that these answer sets are independent of the participants, and thus can be done at the organizers' leisure, and also will be completely re-usable for later research.

Along with the topics and the associated documents (but not the answer sets), the organizers would give participants a small number (2 or 3) of example documents with answers to serve as seeds of clusters for each topic. The evaluation task of the participants would be to find opinion passages in the CD documents and group them with the proper example document answers. Evaluation would be normal recall-precision figures based on overlap with the organizer-constructed answer sets.

An estimate of human resources to conduct such an experiment would be about 6 hours per topic to construct the topics, 2 hours per topic to find a WNC subset (we had a high school student, Ed Slavich, do this), and about 5-7 minutes per document to construct answer segments. This gives a rough estimate of about 1000 hours of time.

2.11   Lessons Learned

- It was anticipated that collection formation would be a substantial obstacle to completion of the project, and it was. There is no real answer to this problem other than get as much out of the way regarding collection as is possible before the workshop actually starts.

- It was recognized early on by everybody that the computing and network resources initially available at MITRE would be insufficient for the medium and large-scale collection related tasks that the workshop need to accomplish, especially for the end-user task. However, the exact needs couldn't be identified until the large collection and methods of securely accessing it had been decided upon. This didn't happen until the end of week 2. With normal bureaucratic delays this meant that the needed hardware was not installed until the middle of week 7, despite massive efforts by several MITRE employees. We should have arranged for more resources before the workshop started, even if we weren't certain what exactly was needed.

- Annotations take a long time, especially when the annotation schemas are still being developed. It probably would have been worthwhile having a longer research break

59

in the middle of the workshop just to allow annotations to progress so that later work dependent on annotations would not be held up.

- In general the MITRE facilities were good. We did not take advantage of the video-conferencing facilities as much as we could have. Those people not physically at the MITRE site were not able to have as much impact on the project as they could have otherwise.

Chapter 3

Catalog of Software, Data, Reports, and Presentations

- The corpus is in /workshops/multip/database at MITRE. The directory structure is documented in /workshops/multip/doc and described above in Chapter 2.

- The conceptual annotation instructions, entitled *Instructions for Annotating Opinions in Newspaper Articles*. Available as a latex/postscript document (annotationInstructions.ps, annotationInstructions.tex).

- *Annotating Opinions in Newspaper Articles: Example Passages with Annotations*. Available as a latex/postscript document (trainingegs.ps,trainingegs.tex).

- *Instructions for Using GATE to Annotate Opinions*. Available as an on-line HTML document at http://www.cs.pitt.edu/mpqa/opinion-annotations/gate-instructions

- Corpus of manually annotated documents. Available at MITRE under the directory /workshops/multip/database. The directory structure is documented in /workshops/multip/doc, and described above in Chapter 2.

- *Conceptulization of Perspective in Language*: Available as a Word document (concept.doc).

- Repository of Linguistic Clues. Available at MITRE under the directory /workshops/multip/lib.

- Learning Architecture. Available at MITRE under the directory /workshops/multip/bin. Documentation for this code can be found in /workshops/multip/doc. In addition, it is described and motivated above in Chapter 2.

- Experimental results. Presented above in Chapter 2.

- SMART retrieval system with clustering tools. Available at MITRE under the directory /workshops/multip/tools/smart/smart.

- Topic collection, including topics, answer segments, and list of documents available through WNC related to topic. Available at MITRE under the directory /workshops/multip/database/topics.

- Midterm Meeting Presentation, June 6, 2002. Available as a PowerPoint presentation (midterm.ppt)

- Final Meeting Presentation, July 22-23, 2002. Available as a PowerPoint presentation (final.ppt)

- A presentation by John Prange, *Thinking about Multiple Perspectives*. Though this is not a technical outcome of the workshop participants, we include it because it provides valuable context for the work. Available as a PowerPoint presentation (jprange.ppt)

References

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

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

Bruce, R. and J. Wiebe. 1999. Recognizing subjectivity: A case study of manual tagging. *Natural Language Engineering*, 5(2).

Cohen, J. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Meas.*, 20:37–46.

Cues and Transitions for the Reader. 2002. from: http://www.mapnp.org/library/writing/cuestran.htm.

Dagan, I., S. Pereira, and Lee. 1994. Similarity-based estimation of word cooccurrence probabilities. In *32th Annual Meeting of the ACL (ACL-94)*, pages 272–278.

Framenet. 2002. *http://www.icsi.berkeley.edu/∼framenet/*.

Grosz, B. and C. Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204.

Hatzivassiloglou, V. and J. Wiebe. 2000. Effects of adjective orientation and gradability on sentence subjectivity. In *18th International Conference on Computational Linguistics (COLING-2000)*.

Hearst, M. 1997. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

Hindle, D. 1990. Noun classification from predicate-argument structures. In *28th Annual Meeting of the ACL (ACL-90)*, pages 268–275.

Hobbs, J. Coherence and coreference. *Cognitive Science*, 3:67–90.

Lee, L. 1999. Measures of distributional similarity. In *Proc. ACL '99*, pages 25–32.

Lee, L. and F. Pereira. 1999. Distributional similarity models: Clustering vs. nearest neighbors. In *Proc. ACL '99*.

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

Lin, D. 1994. Principar–an efficient, broad-coverage, principle-based parser. In *Proc. COLING 94*, pages 482–488.

Lin, D. 1998. Automatic retrieval and clustering of similar words. In *Proc. COLING-ACL '98*, pages 768–773.

Mann, W.C. and S.A. Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

Morris, Jane and Graeme Hirst. 1991. Lexial cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48.

Passonneau, R. and D. Litman. 1993. Intention-based segmentation: Human reliability and correlation with linguistic clues. In *Proc. 31st Annual Meeting of the Association for Computational Linguistics (ACL-93)*, pages 148–155. Association for Computational Linguistics.

Riloff, E. 1995. Little Words Can Make a Big Difference for Text Classification. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 130–136.

Riloff, E. 1996. Automatically Generating Extraction Patterns from Untagged Text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1044–1049. The AAAI Press/MIT Press.

Riloff, E. and R. Jones. 1999. Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*.

Riloff, E. and W. Lehnert. 1994. Information Extraction as a Basis for High-Precision Text Classification. *ACM Transactions on Information Systems*, 12(3):296–333, July.

Thelen, M. and E. Riloff. 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. 2000. Learning subjective adjectives from corpora. In *17th National Conference on Artificial Intelligence (AAAI-2000)*.

Wiebe, J., T. O'Hara, T. Sandgren, and K. McKeever. 1998. An empirical approach to temporal reference resolution. *Journal of Artificial Intelligence Research*, 9:247–293.

Wiebe, J., T. Wilson, and M. Bell. 2001. Identifying collocations for recognizing opinions. In *Proc. ACL-01 Workshop on Collocation: Computational Extraction, Analysis, and Exploitation*, July.

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