Playing the Telephone Game: Determining the Hierarchical Structure of Perspective and Speech Expressions

Eric Breck and Claire Cardie
Department of Computer Science
Cornell University
Ithaca, NY 14853
USA
ebreck,cardie@cs.cornell.edu

Abstract

News articles report on facts, events, and opinions with the intent of conveying the truth. However, the facts, events, and opinions appearing in the text are often known only second- or third-hand, and as any child who has played "telephone" knows, this relaying of facts often garbles the original message. Properly understanding the information filtering structures that govern the interpretation of these facts, then, is critical to appropriately analyzing them. In this work, we present a learning approach that correctly determines the hierarchical structure of information filtering expressions 78.30% of the time.

1 Introduction

Newswire text has long been a primary target for natural language processing (NLP) techniques such as information extraction, summarization, and question answering (e.g. MUC (1998); NIS (2003); DUC (2003)). However, newswire does not offer direct access to facts, events, and opinions; rather, journalists report what they have experienced, and report on the experiences of others. That is, facts, events, and opinions are filtered by the point of view of the writer and other sources. Unfortunately, this filtering of information through multiple sources (and multiple points of view) complicates the natural language interpretation process because the reader (human or machine) must take into account the biases introduced by this indirection. It is important for understanding both newswire and narrative text (Wiebe, 1994), therefore, to appropriately recognize expressions of point of view, and to associate them with their direct and indirect sources.

This paper introduces two kinds of expression that can filter information. First, we define a *perspective expression* to be the minimal span of text that denotes the presence of an explicit opinion, evaluation, emotion, speculation, belief, sentiment, etc.[1] *Private state* is the general term typically used to refer to these mental and emotional states that cannot be directly observed or verified (Quirk et al., 1985). Further, we define the *source* of a perspective expression to be the experiencer of that private state, that is, the person or entity whose opinion or emotion is being conveyed in the text. Second, *speech expressions* simply convey the words of another individual – and by the choice of words, the reporter filters the original source's intent. Consider for example, the following sentences (in which perspective expressions are denoted in **bold**, speech expressions are underlined, and sources are denoted in *italics*):

1. *Charlie* was **angry** at *Alice*'s **claim** that *Bob* was **unhappy**.
2. *Philip Clapp*, president of the National Environment Trust, sums up well the general thrust of the **reaction** of *environmental movements*: "There is no reason at all to believe that the polluters are suddenly going to become reasonable."

Perspective expressions in Sentence 1 describe the emotions or opinion of three sources: Charlie's anger, Bob's unhappiness, and Alice's belief. Perspective expressions in Sentence 2, on the other hand, introduce the explicit opinion of one source, i.e. the reaction of the environmental movements. Speech expressions also perform filtering in these examples. The reaction of the environmental movements is filtered by Clapp's summarization, which, in turn, is filtered by the writer's choice of quotation. In addition, the fact that Bob was unhappy is filtered through Alice's claim, which, in turn, is filtered by the writer's choice of words for the sentence. Similarly, it is only according to the writer that Charlie is angry.

The specific goal of the research described here is to accurately identify the hierarchical structure of *perspective and speech expressions* (pse's) in text.[2]

[1] Note that *implicit* expressions of perspective, i.e. Wiebe et al.'s (2003) "expressive subjective elements" are not the subject of study here.

[2] For the rest of this paper, then, we ignore the distinction between perspective and speech expressions, so in future ex-

Given sentences 1 and 2 and their pse's, for example, we will present methods that produce the structures shown in Figure 1, which represent the multi-stage information filtering that should be taken into account in the interpretation of the text.

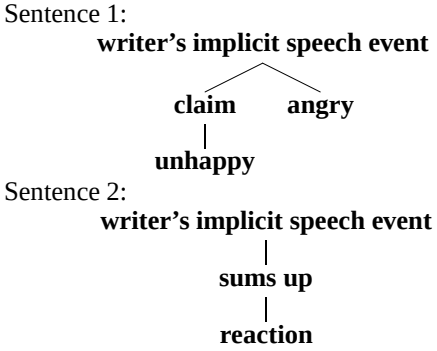

Figure 1: Hierarchical structure of the perspective and speech expressions in sentences 1 and 2

We propose a supervised machine learning approach to the problem that relies on a small set of syntactically-based features. More specifically, the method first trains a binary classifier to make pairwise parent-child decisions among the pse's in the same sentence, and then combines the decisions to determine their global hierarchical structure. We compare the approach to two heuristic-based baselines — one that simply assumes that every pse is filtered only through the writer, and a second that is based on syntactic dominance relations in the associated parse tree. In an evaluation using the opinion-annotated NRRC corpus (Wiebe et al., 2002), the learning-based approach achieves an accuracy of 78.30%, significantly higher than both the simple baseline approach (65.57%) and the parse-based baseline (71.64%). We believe that this study provides a first step towards understanding the multi-stage filtering process that can bias and garble the information present in newswire text.

The rest of the paper is organized as follows. We present related work in Section 2 and describe the machine learning approach in Section 3. The experimental methodology and results are presented in Sections 4 and 5, respectively. Section 6 summarizes our conclusions and plans for future work.

2 The Larger Problem and Related Work

This paper addresses the problem of identifying the hierarchical structure of perspective and speech expressions. We view this as a necessary and important component of a larger perspective-analysis system. Such a system would be able to identify all pse's in a document, as well as identify their structure. The system would also identify the direct source of each pse. Finally, the system would identify the text corresponding to the content of a private state or the speech expressed by a pse.[3] Such a system might analyze sentence 2 as follows:

(source: *writer*
pse: **(implicit speech event)**
content: Philip ... reasonable.")
|
(source: *clapp*
pse: **sums up**
content: "There ... reasonable.")
|
(source: *environmental movements*
pse: **reaction**
content: (no text))

As far as we are aware, no single system exists that simultaneously solves all these problems. There is, however, quite a bit of work that addresses various pieces of this larger task, which we will now survey.

Gerard (2000) proposes a computational model of the reader of a news article. Her model provides for multiple levels of hierarchical beliefs, such as the nesting of a primary source's belief within that of a reporter. However, Gerard does not provide algorithms for extracting this structure directly from newswire texts.

Bethard et al. (2004) seek to extract propositional

pse class	count
writer	9808
verb	7623
noun	2293
no parse	278
adjective	197
adverb	50
other	370

Table 1: Breakdown of classes of pse's. "writer" denotes pse's with the writer as source. "No parse" denotes pse's in sentences where the parse failed, and so the part of speech could not be determined.

number of pse's	number of sentences
1	3612
2	3256
3	1810
4	778
5	239
>5	113

Table 2: Breakdown of number of pse's per sentence

amples, both types of pse appear in boldface. Note that the acronym 'pse' has been used previously with a different meaning (Wiebe, 1994).

[3] In (Wiebe, 2002), this is referred to as the *inside*.

opinions and their holders. They define an opinion as "a sentence, or part of a sentence that would answer the question 'How does X feel about Y?' " A propositional opinion is an opinion "localized in the propositional argument" of certain verbs, such as "believe" or "realize". Their task then corresponds to identifying a pse, its associated direct source, and the content of the private state. However, they consider as pse's only verbs, and further restrict attention to verbs with a propositional argument, which is a subset of the perspective and speech expressions that we consider here. Table 1, for example, shows the diversity of word classes that correspond to pse's in our corpus. Perhaps more importantly for the purposes of this paper, their work does not address information filtering issues, i.e. problems that arise when an opinion has been filtered through multiple sources. Namely, Bethard et al. (2004) do not consider sentences that contain multiple pse's, and do not, therefore, need to identify any indirect sources of opinions. As shown in Table 2, however, we find that sentences with multiple non-writer pse's (i.e. sentences that contain 3 or more total pse's) comprise a significant portion (29.98%) of our corpus. An advantage over our work, however, is that Bethard et al. (2004) do not require separate solutions to pse identification and the identification of their direct sources.

Automatic identification of sources has also been addressed indirectly by Gildea and Jurafsky's (2002) work on semantic role identification in that finding sources often corresponds to finding the filler of the agent role for verbs. Their methods then might be used to identify sources and associate them with pse's that are verbs or portions of verb phrases. Whether their work will also apply to pse's that are realized as other parts of speech is an open question.

Wiebe (1994), studies methods to track the change of "point of view" in narrative text (fiction). That is, the "writer" of one sentence may not correspond to the writer of the next sentence. Although this is not as frequent in newswire text as in fiction, it will still need to be addressed in a solution to the larger problem.

Bergler (1993) examines the lexical semantics of speech event verbs in the context of generative lexicon theory. While not specifically addressing our problem, the "semantic dimensions" of reporting verbs that she extracts might be very useful as features in our approach.

Finally, Wiebe et al. (2003) present preliminary results for the automatic identification of perspective and speech expressions using corpus-based techniques. While the results are promising (66% F-measure), the problem is still clearly unsolved. As explained below, we will instead rely on manually tagged pse's for the studies presented here.

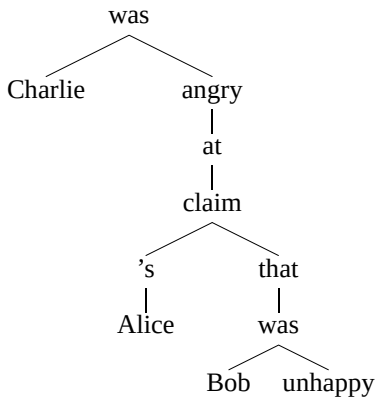

Figure 2: Dependency parse of sentence 1 according to the Collins parser.

3 The Approach

Our task is to find the hierarchical structure among the pse's in individual sentences. One's first impression might be that this structure should be obvious from the syntax: one pse should filter another roughly when it dominates the other in a dependency parse. This heuristic, for example, would succeed for "claim" and "unhappy" in sentence 1, whose pse structure is given in Figure 1 and parse structure (as produced by the Collins parser) in Figure 2. [4]

Even in sentence 1, though, we can see that the problem is more complex: "angry" dominates "claim" in the parse tree, but does not filter it. Unfortunately, an analysis of the parse-based heuristic on our training data (the data set will be described in Section 4), uncovered numerous, rather than just a few, sources of error. Therefore, rather than trying to handcraft a more complex collection of heuristics, we chose to adopt a supervised machine learning approach that relies on features identified in this analysis. In particular, we will first train a binary classifier to make pairwise decisions as to whether a given pse is the immediate parent of another. We then use a simple approach to combine these decisions to find the hierarchical information-filtering structure of all pse's in a sentence.

We assume that we have a training corpus of

[4] For this heuristic and the features that follow, we will speak of the pse's as if they had a position in the parse tree. However, since pse's are often multiple words, and do not necessarily form a constituent, this is not entirely accurate. The parse node corresponding to a pse will be the highest node in the dependency parse corresponding to a word in the pse. We consider the writer's implicit pse to correspond to the root of the parse.

sentences, annotated with pse's and their hierarchical pse structure (Section 4 describes the corpus). Training instances for the binary classifier are pairs of pse's from the same sentence, $\langle pse_{target}, pse_{parent} \rangle$[5]. We assign a class value of 1 to a training instance if pse_{parent} is the immediate parent of pse_{target} in the manually annotated hierarchical structure for the sentence, and 0 otherwise. For sentence 1, there are nine training instances generated: $\langle$claim, writer$\rangle$, $\langle$angry, writer$\rangle$, $\langle$unhappy, claim$\rangle$ (class 1), $\langle$claim, angry$\rangle$, $\langle$claim, unhappy$\rangle$, $\langle$angry, claim$\rangle$, $\langle$angry, unhappy$\rangle$, $\langle$unhappy, writer$\rangle$, $\langle$unhappy, angry$\rangle$ (class 0). The features used to describe each training instance are explained below.

During testing, we construct the hierarchical pse structure of an entire sentence as follows. For each pse in the sentence, ask the binary classifier to judge each other pse as a potential parent, and choose the pse with the highest confidence[6]. Finally, join these immediate-parent links to form a tree.[7]

One might also try comparing pairs of potential parents for a given pse, or other more direct means of ranking potential parents. We chose what seemed to be the simplest method for this first attempt at the problem.

3.1 Features

Here we motivate and describe the 23 features used in our model. Unless otherwise stated, all features are binary (1 if the described condition is true, 0 otherwise).

Parse-based features (6). Based on the performance of the parse-based heuristic, we include a pse_{parent}-dominates-pse_{target} feature in our feature set. To compensate for parse errors, however, we also include a variant of this that is 1 if the parent of pse_{parent} dominates pse_{target}.

Many filtering expressions filter pse's that occur in their complements, but not in adjuncts. Therefore, we add variants of the previous two syntax-based features that denote whether the parent node dominates pse_{target}, but only if the first dependency relation is an object relation.

For similar reasons, we include a feature calculating the domination relation based on a partial parse. Consider the following sentence:

3. He was **criticized** more than **recognized** for his policy.

One of "criticized" or "recognized" will be the root of this dependency parse, thus dominating the other, and suggesting (incorrectly) that it filters the other pse. Because a partial parse does not attach all constituents, such spurious dominations are eliminated. The partial parse feature is 1 for fewer instances than pse_{parent}-dominates-pse_{target}, but it is more indicative of a positive instance when it is 1.

So that the model can adjust when the parse is not present, we include a feature that is 1 for all instances generated from sentences on which the parser failed.

Positional features (5). Forcing the model to decide whether pse_{parent} is the parent of pse_{target} without knowledge of the other pse's in the sentence is somewhat artificial. We therefore include several features that encode the relative position of pse_{parent} and pse_{target} in the sentence. Specifically, we add a feature that is 1 if pse_{parent} is the root of the parse (and similarly for pse_{target}). We also include a feature giving the ordinal position of pse_{parent} among the pse's in the sentence, relative to pse_{target} (-1 means pse_{parent} is the pse that immediately precedes pse_{target}, 1 means immediately following, and so forth). To allow the model to vary when there are more potential parents to choose from, we include a feature giving the total number of pse's in the sentence.

Special parents and lexical features (6). Some particular pse's are special, so we specify indicator features for four types of parents: the writer pse, and the lexical items "said" (the most common non-writer pse) and "according to". "According to" is special because it is generally not very high in the parse, but semantically tends to filter everything else in the sentence.

In addition, we include as features the part of speech of pse_{parent} and pse_{target} (reduced to noun, verb, adjective, adverb, or other), since intuitively we expected distinct parts of speech to behave differently in their filtering.

Genre-specific features (6). Finally, journalistic writing contains a few special forms that are not always parsed accurately. Examples are:

4. "Alice **disagrees** with me," Bob **argued**.

[5] We skip sentences where there is no decision to make (sentences with zero or one non-writer pse). Since the writer pse is the root of every structure, we do not generate instances with the writer pse in the pse_{target} position.

[6] There is an ambiguity if the classifier assigns the same confidence to two potential parents. For evaluation purposes, we consider the classifier's response incorrect if any of the highest-scoring potential parents are incorrect.

[7] The directed graph resulting from flawed automatic predictions might not be a tree (i.e. it might be cyclic and disconnected). Since this occurs very rarely (5 out of 9808 sentences on the test data), we do not attempt to correct any non-tree graphs.

5. Charlie, she **noted**, **dislikes** Chinese food.

The parser may not recognize that "noted" and "argued" should dominate all other pse's in sentences 4 and 5, so we attempt to recognize when a sentence falls into one of these two patterns. For $\langle \text{disagrees}, \text{argued} \rangle$ generated from sentence 4, features pse_{parent}-pattern-1 and pse_{target}-pattern-1 would be 1, while for $\langle \text{dislikes}, \text{noted} \rangle$ generated from sentence 5, feature pse_{parent}-pattern-2 would be 1. We also add features that denote whether the pse in question falls between matching quote marks. Finally, a simple feature indicates whether pse_{parent} is the last word in the sentence.

3.2 Resources

We rely on a variety of resources to generate our features. The corpus (see Section 4) is distributed with annotations for sentence breaks, tokenization, and part of speech information automatically generated by the GATE toolkit (Cunningham et al., 2002).[8] For parsing we use the Collins (1999) parser.[9] For partial parses, we employ CASS (Abney, 1997). Finally, we use a simple finite-state recognizer to identify (possibly nested) quoted phrases.

For classifier construction, we use the IND package (Buntine, 1993) to train decision trees (we use the `mml` tree style, a minimum message length criterion with Bayesian smoothing).

4 Data Description

The data for these experiments come from version 1.1 of the NRRC corpus (Wiebe et al., 2002).[10]. The corpus consists of 535 newswire documents (mostly from the FBIS), of which we used 66 (1375 sentences) for developing the heuristics and features, while keeping the remaining 469 (9808 sentences) blind (used for 10-fold cross-validation).

Although the NRRC corpus provides annotations for all pse's, it does not provide annotations to denote directly their hierarchical structure within a sentence. This structure must be extracted from an attribute of each pse annotation, which lists the pse's direct and indirect sources. For example, the "source chain" for "unhappy" in sentence 1, would be (*writer*, *Alice*, *Bob*). The source chains allow us to automatically recover the hierarchical structure of the pse's: the parent of a pse with source chain $(s_0, s_1, \ldots s_{n-1}, s_n)$ is the pse with source chain $(s_0, s_1, \ldots s_{n-1})$. Unfortunately, ambiguities can arise. Consider the following sentence:

6. *Bob* **said**, "you're welcome" because *he* **was glad** to see that *Mary* **was happy**.

Both "said" and "was glad" have the source chain (*writer*, *Bob*),[11] while "was happy" has the source chain (*writer*, *Bob*, *Mary*). It is therefore not clear from the manual annotations whether "was happy" should have "was glad" or "said" as its parent. 5.82% of the pse's have ambiguous parentage (i.e. the recovery step finds a set of parents $P(pse)$ with $|P(pse)| > 1$). For training, we assign a class value of 1 to all instances $\langle pse, par \rangle, par \in P(pse)$. For testing, if an algorithm attaches pse to any element of $P(pse)$, we score the link as correct (see Section 5.1). Since ultimately our goal is to find the sources through which information is filtered (rather than the pse's), we believe this is justified.

For training and testing, we used only those sentences that contain at least two non-writer pse's[12] – for all other sentences, there is only one way to construct the hierarchical structure. Again, Table 2 presents a breakdown (for the test set) of the number of pse's per sentence – thus we only use approximately one-third of all the sentences in the corpus.

5 Results and Discussion

5.1 Evaluation

How do we evaluate the performance of an automatic method of determining the hierarchical structure of pse's? Lin (1995) proposes a method for evaluating dependency parses: the score for a sentence is the fraction of correct parent links identified; the score for the corpus is the average sentence score. Formally, the score for a

[8] GATE's sentences sometimes extend across paragraph boundaries, which seems never to be warranted. Inaccurately joining sentences has the effect of adding more noise to our problem, so we split GATE's sentences at paragraph boundaries, and introduce writer pse's for the newly created sentences.

[9] We convert the parse to a dependency format that makes some of our features simpler using a method similar to the one described in Xia and Palmer (2001). We also employ a method from Adam Lopez at the University of Maryland to find grammatical relationships between words (subject, object, etc.).

[10] The original corpus is available at `http://nrrc.mitre.org/NRRC/Docs_Data/MPQA_04/approval_mpqa.htm`. Code and data used in our experiments are available at `http://www.cs.cornell.edu/~ebreck/breck04playing/`.

[11] The annotators also performed coreference resolution on sources.

[12] Under certain circumstances, such as paragraph-long quotes, the writer of a sentence will not be the same as the writer of a document. In such sentences, the NRRC corpus contains additional pse's for any other sources besides the writer of the document. Since we are concerned in this work only with one sentence at a time, we discard all such implicit pse's besides the writer of the sentence. Also, in a few cases, more than one pse in a sentence was marked as having the writer as its source. We believe this to be an error and so discarded all but one writer pse.

metric	size	heurOne	heurTwo	decTree
Lin	2940	65.57%	71.64%	78.30%
perf	2940	36.02%	45.37%	54.52%
bin	21933	73.20%	77.73%	82.12%
bin +	7882	60.63%	66.94%	70.35%
bin −	14051	80.24%	83.78%	88.72%

Table 3: Performance on test data. "Lin" is Lin's dependency score, "perf" is the fraction of sentences whose structure was identified perfectly, and "bin" is the performance of the binary classifier (broken down for positive and negative instances). "Size" is the number of sentences or pse pairs.

# pse's	# sents	heurOne	heurTwo	decTree
3	1810	70.88%	75.41%	81.82%
4	778	59.17%	67.82%	74.38%
5	239	53.87%	61.92%	68.93%
>5	113	49.31%	58.03%	68.68%

Table 4: Performance by number of pse's per sentence

method evaluated on the entire corpus ("Lin") is $\frac{\sum_{s \in S} \frac{|\{pse | pse \in Non\_writer\_pse's(s) \wedge parent(pse) = autopar(pse))\}|}{|Non\_writer\_pse's(s)|}}{|S|}$, where S is the set of all sentences in the corpus, $Non\_writer\_pse's(s)$ is the set of non-writer pse's in sentence s, $parent(pse)$ is the correct parent of pse, and $autopar(pse)$ is the automatically identified parent of pse.

We also present results using two other (related) metrics. The "perf" metric measures the fraction of sentences whose structure is determined entirely correctly (i.e. "perf"ectly). "Bin" is the accuracy of the binary classifier (with a 0.5 threshold) on the instances created from the test corpus. We also report the performance on positive and negative instances.

5.2 Results

We compare the learning-based approach (*decTree*) to the heuristic-based approaches introduced in Section 3 — *heurOne* assumes that all pse's are attached to the writer's implicit pse; *heurTwo* is the parse-based heuristic that relies solely on the dominance relation[13].

We use 10-fold cross-validation on the evaluation data to generate training and test data (although the heuristics, of course, do not require training). The results of the decision tree method and the two heuristics are presented in Table 3.

[13]That is, heurTwo attaches a pse to the pse most immediately dominating it in the dependency tree. If no other pse dominates it, a pse is attached to the writer's pse.

5.3 Discussion

Encouragingly, our machine learning method uniformly and significantly[14] outperforms the two heuristic methods, on all metrics and in sentences with any number of pse's. The difference is most striking in the "perf" metric, which is perhaps the most intuitive. Also, the syntax-based heuristic (heurTwo) significantly[15] outperforms heurOne, confirming our intuitions that syntax is important in this task.

As the binary classifer sees many more negative instances than positive, it is unsurprising that its performance is much better on negative instances. This suggests that we might benefit from machine learning methods for dealing with unbalanced datasets.

Examining the errors of the machine learning system on the development set, we see that for half of the pse's with erroneously identified parents, the parent is either the writer's pse, or a pse like "said" in sentences 4 and 5 having scope over the entire sentence. For example,

7. "Our **concern** is whether persons used to the role of policy implementors can objectively **assess and critique** executive policies which impinge on human rights," **said** Ramdas.

Our model chose the parent of "assess and critique" to be "said" rather than "concern." We also see from Table 4 that the model performs more poorly on sentences with more pse's. We believe that this reflects a weakness in our decision to combine binary decisions, because the model has learned that in general, a "said" or writer's pse (near the root of the structure) is likely to be the parent, while it sees many fewer examples of pse's such as "concern" that lie in the middle of the tree.

Although we have ignored the distinction throughout this paper, error analysis suggests speech event pse's behave differently than private state pse's with respect to how closely syntax reflects their hierarchical structure. It may behoove us to add features to allow the model to take this into account. Other sources of error include erroneous sentence boundary detection, parenthetical statements (which the parser does not treat correctly for our purposes) and other parse errors, partial quotations, as well as some errors in the annotation.

Examining the learned trees is difficult because of their size, but looking at one tree to depth three

[14]$p < 0.01$, using an approximate randomization test with 9,999 trials. See (Eisner, 1996, page 17) and (Chinchor et al., 1993, pages 430-433) for descriptions of this method.

[15]Using the same test as above, $p < 0.01$, except for the performance on sentences with more than 5 pse's, because of the small amount of data, where $p < 0.02$.

reveals a fairly intuitive model. Ignoring the probabilities, the tree decides pse_{parent} is the parent of pse_{target} if and only if pse_{parent} is the writer's pse (and pse_{target} is not in quotation marks), or if pse_{parent} is the word "said." For all the trees learned, the root feature was either the writer pse test or the partial-parse-based domination feature.

6 Conclusions and Future Work

We have presented the concept of perspective and speech expressions, and argued that determining their hierarchical structure is important for natural language understanding of perspective. We have shown that identifying the hierarchical structure of pse's is amenable to automated analysis via a machine learning approach, although there is room for improvement in the results.

In the future, we plan to address the related tasks discussed in Section 2, especially identifying pse's and their immediate sources. We are also interested in ways of improving the machine learning formulation of the current task, such as optimizing the binary classifier on the whole-sentence evaluation, or defining a different binary task that is easier to learn. Nevertheless, we believe that our results provide a step towards the development of natural language systems that can extract and summarize the viewpoints and perspectives expressed in text while taking into account the multi-stage information filtering process that can mislead more naïve systems.

Acknowledgments

This work was supported in part by NSF Grant IIS-0208028 and by an NSF Graduate Research Fellowship. We thank Rebecca Hwa for creating the dependency parses. We also thank the Cornell NLP group for helpful suggestions on drafts of this paper. Finally, we thank Janyce Wiebe and Theresa Wilson for draft suggestions and advice regarding this problem and the NRRC corpus.

References

Steven Abney. 1997. The SCOL manual. cass is available from http://www.vinartus.net/spa/scol1h.tar.gz.

Sabine Bergler. 1993. Semantic dimensions in the field of reporting verbs. In *Proceedings of the Ninth Annual Conference of the University of Waterloo Centre for the New Oxford English Dictionary and Text Research*, Oxford, England, September.

Steven Bethard, Hong Yu, Ashley Thornton, Vasileios Hatzivassiloglou, and Dan Jurafsky. 2004. Automatic extraction of opinion propositions and their holders. In *Working Notes of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*. March 22-24, 2004, Stanford.

Wray Buntine. 1993. Learning classification trees. In D. J. Hand, editor, *Artificial Intelligence frontiers in statistics*, pages 182–201. Chapman & Hall,London. Available at http://ic.arc.nasa.gov/projects/bayes-group/ind/IND-program.html.

Nancy Chinchor, Lynette Hirschman, and David Lewis. 1993. Evaluating message understanding systems: An analysis of the third message understanding conference (MUC-3). *Computational Linguistics*, 19(3):409–450.

Michael John Collins. 1999. *Head-driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania, Philadelphia.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. GATE: A framework and graphical development environment for robust nlp tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL '02)*, Philadelphia, July.

2003. *Proceedings of the Workshop on Text Summarization*, Edmonton, Alberta, Canada, May. Presented at the 2003 Human Language Technology Conference.

Jason Eisner. 1996. An empirical comparison of probability models for dependency grammar. Technical Report IRCS-96-11, IRCS, University of Pennsylvania.

Christine Gerard. 2000. Modelling readers of news articles using nested beliefs. Master's thesis, Concordia University, Montréal, Québec, Canada.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Dekang Lin. 1995. A dependency-based method for evaluating broad-coverage parsers. In *IJCAI*, pages 1420–1427.

1998. *Proceedings of the Seventh Message Understanding Conference (MUC-7)*. Morgan Kaufman, April.

NIST. 2003. *Proceedings of The Twelfth Text REtrieval Conference (TREC 2003)*, Gaithersburg, MD, November. NIST special publication SP 500-255.

Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1985. *A Comprehensive Grammar of the English Language*. Longman, New York.

J. Wiebe, E. Breck, C. Buckley, C. Cardie, P. Davis, B. Fraser, D. Litman, D. Pierce, E. Riloff, and T. Wilson. 2002. NRRC Summer Workshop on Multiple-Perspective Question Answering Final Report. Tech report, NRRC, Bedford, MA.

J. Wiebe, E. Breck, C. Buckley, C. Cardie, P. Davis, B. Fraser, D. Litman, D. Pierce, E. Riloff, T. Wilson, D. Day, and M. Maybury. 2003. Recognizing and Organizing Opinions Expressed in the World Press. In *Papers from the AAAI Spring Symposium on New Directions in Question Answering (AAAI tech report SS-03-07)*. March 24-26, 2003. Stanford.

Janyce Wiebe. 1994. Tracking point of view in narrative. *Computational Linguistics*, 20(2):233–287.

Janyce Wiebe. 2002. Instructions for annotating opinions in newspaper articles. Technical Report TR-02-101, Dept. of Comp. Sci., University of Pittsburgh.

Fei Xia and Martha Palmer. 2001. Converting dependency structures to phrase structures. In *Proc. of the HLT Conference*.