

OPINE: Extracting Product Features and Opinions from Reviews

Ana-Maria Popescu

Bao Nguyen

Oren Etzioni

Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350
{amp,omicron,etzioni}@cs.washington.edu

Abstract

Consumers have to often wade through a large number of on-line reviews in order to make an informed product choice. We introduce OPINE, an unsupervised, high-precision information extraction system which mines product reviews in order to build a model of product features and their evaluation by reviewers.

1 Introduction

The Web contains a wealth of customer reviews - as a result, the problem of “review mining” has seen increasing attention over the last few years from (Turney, 2003; Hu and Liu, 2004) and many others. We decompose the problem of review mining into the following subtasks: **a) Identify product features, b) Identify opinions regarding product features, c) Determine the polarity of each opinion and d) Rank opinions according to their strength** (e.g., “abominable” is stronger than “bad”).

We introduce OPINE, an unsupervised information extraction system that embodies a solution to each of the above subtasks. The remainder of this paper is organized as follows: Section 2 describes OPINE’s components together with their experimental evaluation and Section 3 describes the related work.

2 OPINE Overview

OPINE is built on top of KNOWITALL, a Web-based, domain-independent information extraction system (Etzioni et al., 2005). Given a set of relations of interest, KNOWITALL instantiates relation-specific generic extraction patterns into extraction rules which find candidate facts. The Assessor module then assigns a probability to each candidate using a form of *Point-wise Mutual Information* (PMI) between phrases that is estimated from Web search engine hit counts (Turney, 2003). It

Input: product class C , reviews R .

Output: set of [feature, ranked opinion list] tuples

```
R' ← parseReviews(R);
E ← findExplicitFeatures(R', C);
O ← findOpinions(R', E);
CO ← clusterOpinions(O);
I ← findImplicitFeatures(CO, E);
RO ← rankOpinions(CO);
{(f, oi, ...oj)} ← outputTuples(RO, IUE);
```

Figure 1: OPINE Overview.

computes the PMI between each fact and *discriminator phrases* (e.g., “is a scanner” for the `isa()` relationship in the context of the `Scanner` class). Given fact f and discriminator d , the computed PMI score is:

$$\text{PMI}(f, d) = \frac{\text{Hits}(d + f)}{\text{Hits}(d) * \text{Hits}(f)}$$

The PMI scores are converted to binary features for a Naive Bayes Classifier, which outputs a probability associated with each fact.

Given product class C with instances I and reviews R , OPINE’s goal is to find the set of (feature, opinions) tuples $\{(f, o_i, \dots o_j)\}$ s.t. $f \in F$ and $o_i, \dots o_j \in O$, where:

- F is the set of product class features in R .
- O is the set of opinion phrases in R .
- opinions associated with a particular feature are ranked based on their strength.

OPINE’s solution to this task is outlined in Figure 1. In the following, we describe in detail each step.

Explicit Feature Extraction OPINE parses the reviews using the MINIPAR dependency parser (Lin, 1998) and applies a simple pronoun-resolution module to the parsed data. The system then finds *explicitly mentioned* product features (E) using an extended version of KNOWITALL’s extract-and-assess strategy described above. OPINE extracts the following types of *product features*: properties, parts, features of product parts (e.g., `ScannerCoverSize`), *related concepts* (e.g., `Image`

is related to Scanner) and parts and properties of related concepts (e.g., ImageSize). When compared on this task with the most relevant previous review-mining system in (Hu and Liu, 2004), OPINE obtains a 22% improvement in precision with only a 3% reduction in recall on the relevant 5 datasets. One third of this increase is due to OPINE’s feature assessment step and the rest is due to the use of Web PMI statistics.

Opinion Phrases OPINE extracts adjective, noun, verb and adverb phrases attached to explicit features as potential opinion phrases. OPINE then collectively assigns *positive*, *negative* or *neutral* semantic orientation (SO) labels to their respective head words. This problem is similar to labeling problems in computer vision and OPINE uses a well-known computer vision technique, *relaxation labeling*, as the basis of a 3-step SO label assignment procedure. First, OPINE identifies the average SO label for a word w in the context of the review set. Second, OPINE identifies the average SO label for each word w in the context of a feature f and of the review set (“hot” has a negative connotation in “hot room”, but a positive one in “hot water”). Finally, OPINE identifies the SO label of word w in the context of feature f and sentence s . For example, some people like large scanners (“I love this large scanner”) and some do not (“I hate this large scanner”). The phrases with non-neutral head words are retained as opinion phrases and their polarity is established accordingly. On the task of opinion phrase extraction, OPINE obtains a precision of 79% and a recall of 76% and on the task of opinion phrase polarity extraction OPINE obtains a precision of 86% and a recall of 84%.

Implicit Features Opinion phrases refer to *properties*, which are sometimes *implicit* (e.g., “tiny phone” refers to the phone size). In order to extract such properties, OPINE first clusters opinion phrases (e.g., *tiny* and *small* will be placed in the same cluster), automatically labels the clusters with property names (e.g., *Size*) and uses them to build *implicit* features (e.g., *PhoneSize*). Opinion phrases are clustered using a mixture of WordNet information (e.g., antonyms are placed in the same cluster) and lexical pattern information (e.g., “clean, almost spotless” suggests that “clean” and “spotless” are likely to refer to the same property). (Hu and Liu, 2004) doesn’t handle implicit features, so we have evaluated the impact of implicit feature extraction on two separate sets of reviews in the Hotels and Scanners domains. Extracting implicit features (in addition to explicit features) has resulted in a 2% increase in precision and a 6% increase in recall for OPINE on the task of feature extraction.

Ranking Opinion Phrases Given an opinion cluster, OPINE uses the final probabilities associated with the SO labels in order to derive an initial opinion phrase strength ranking (e.g., *great* > *good* > *average*) in the manner of (Turney, 2003). OPINE then uses Web-derived con-

straints on the relative strength of phrases in order to improve this ranking. Patterns such as “ a_1 , (*) even a_2 ” are good indicators of how strong a_1 is relative to a_2 . OPINE bootstraps a set of such patterns and instantiates them with pairs of opinions in order to derive constraints such as $strength(deafening) > strength(loud)$. OPINE also uses synonymy and antonymy-based constraints such as $strength(clean) = strength(dirty)$. The constraint set induces a constraint satisfaction problem whose solution is a ranking of the respective cluster opinions (the remaining opinions maintain their default ranking). OPINE’s accuracy on the opinion ranking task is 87%. Finally, OPINE outputs a set of (feature, ranked opinions) tuples for each product.

3 Related Work

The previous review-mining systems most relevant to our work are (Hu and Liu, 2004) and (Kobayashi et al., 2004). The former’s precision on the explicit feature extraction task is 22% lower than OPINE’s while the latter employs an iterative semi-automatic approach which requires significant human input; neither handles *implicit* features. Unlike previous research on identifying the subjective character and the polarity of phrases and sentences ((Hatzivassiloglou and Wiebe, 2000; Turney, 2003) and many others), OPINE identifies the context-sensitive polarity of opinion phrases. In contrast to supervised methods which distinguish among strength levels for sentences or clauses ((Wilson et al., 2004) and others), OPINE uses an unsupervised constraint-based opinion ranking approach.

References

- O. Etzioni, M. Cafarella, D. Downey, S. Kok, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134.
- V. Hatzivassiloglou and J. Wiebe. 2000. Effects of Adjective Orientation and Gradability on Sentence Subjectivity. In *COLING*, pages 299–305.
- M. Hu and B. Liu. 2004. Mining and Summarizing Customer Reviews. In *KDD*, pages 168–177, Seattle, WA.
- N. Kobayashi, K. Inui, K. Tateishi, and T. Fukushima. 2004. Collecting Evaluative Expressions for Opinion Extraction. In *IJCNLP*, pages 596–605.
- D. Lin. 1998. Dependency-based evaluation of MINIPAR. In *Workshop on Evaluation of Parsing Systems at ICLRE*.
- P. Turney. 2003. Inference of Semantic Orientation from Association. In *CoRR cs. CL/0309034*.
- T. Wilson, J. Wiebe, and R. Hwa. 2004. Just how mad are you? finding strong and weak opinion clauses. In *AAAI*, pages 761–769.