Sentiment Extraction: Integrating Statistical Parsing, Semantic Analysis, and Common Sense Reasoning

Lokendra Shastri, Anju G. Parvathy, Abhishek Kumar, John Wesley, and **Rajesh Balakrishnan**
Center for Knowledge Driven Information Systems, Software Engineering and Technology Labs
Infosys Technologies Ltd., Bangalore, India
{*lokendra_shastri, anjug_parvathy, abhishek_kumar25, john_wesley, rajeshb*}*@infosys.com*

Abstract

Much of the ongoing explosion of digital content is in the form of text. This content is a virtual gold-mine of information that can inform a range of social, governmental, and business decisions. For example, using content available on blogs and social networking sites businesses can find out what its customers are saying about their products and services. In the digital age where customer is king, the business value of ascertaining consumer sentiment cannot be overstated. People express sentiments in myriad ways. At times, they use simple, direct assertions, but most often they use sentences involving comparisons, conjunctions expressing multiple and possibly opposing sentiments about multiple features and entities, and pronominal references whose resolution requires discourse level context. Frequently people use abbreviations, slang, SMSese, idioms and metaphors. Understanding the latter also requires common sense reasoning. In this paper, we present *iSEE*, a fully implemented sentiment extraction engine, which makes use of statistical methods, classical NLU techniques, common sense reasoning, and probabilistic inference to extract entity and feature specific sentiment from complex sentences and dialog. Most of the components of *iSEE* are domain independent and the system can be generalized to new domains by simply adding domain relevant lexicons.

1 Introduction

Much of the ongoing explosion of digital content is in the form of text and appears in news articles, blogs, tweets, social networking sites, and resources such as the Wikipedia. This content is a virtual gold-mine of information that can inform a range of social, governmental, and business decisions. For example, businesses can gather intelligence about their competitors (which company is acquiring which company, and who is hiring whom), or find out what its customers are saying about their products and services and about those of its competitors. Businesses can use this information to improve existing offerings, design better products, and address customer concerns. Customers can follow the buzz and find out what other customers are saying about products and services. The latter examples fall in the category of "sentiment analysis" wherein the task involves examining blogs, service call logs, reviews, and postings in social network sites in order to ascertain the opinions (or sentiment) being expressed about products and services. In the digital age where customer is king, the business value of ascertaining consumer sentiment cannot be overstated.

Sentiment analysis (SA) has many variants. In some cases, the task involves processing a collection of postings and arriving at a holistic rating indicating the strength of positive (and negative) sentiments expressed in a blog. In other cases, the analysis is carried out at a much finer granularity whereby each blogger's opinion about a specific product, or even a specific feature of a product, is extracted (What is Joe124's sentiment about the download speed of AcmeNetworks' DSL offering).

In the context of Natural Language Understanding (NLU), Sentiment Analysis (SA) lies at an interesting point on the "level of difficulty" scale. On the one hand, the final outcome (or meaning) of *SA* is straightforward; it consists of (i) a valence (positive/negative), (ii) a magnitude (with a small number of distinct values) and (iii) a specification of the entity about which the sentiment is expressed. But on the other hand, the input of SA can be arbitrarily complex. A cursory review of blog posts would reveal that people express sentiments in myriad ways. At times, they use simple, direct assertions (e.g., I love AcmeNetwork's customer service), but most often they use sentences involving comparisons (The download speed of AcmeNetworks is better than that of AceNet); conjunctions that express multiple and possibly opposing sentiments about multiple features and entities; and pronominal references whose resolution requires discourse level context. Frequently people use abbreviations, slang, SMSese, idioms (AceNet downloads happen at a snail's pace) and metaphors - even productive ones (Downloading on AceNet is like pouring honey from a jar). The understanding of such productive metaphors requires common sense reasoning. Finally, people often resort to sarcasm, irony and humor, all of which pose even greater challenges.

Much progress has been made in the field of text-mining and sentiment analysis (see Section 2 for citations). Some of the most impressive work has been done using statistical techniques based on machine-learning or word-correlation algorithms. These techniques have provided fairly good performance and they also offer the added advantage of being scalable and amenable to automation. While the overall per-

formance of these techniques is surprisingly good, they are less effective in teasing apart multiple sentiments being expressed about multiple entities in a single sentence. They also have difficulty in dealing with discourse-level context (e.g., resolving pronominal references) and understanding the meaning of productive metaphors that require common sense reasoning. An alternative approach to text analytics and sentiment analysis is based on NLU where a text undergoes syntactic, semantic and discourse analysis in order to extract "meaning" expressible in some formal notation. The major shortcoming of this approach is that it is labor intensive, and hence, difficult to scale. As one might expect, researchers have developed hybrid approaches that combine statistical techniques with NLU leading to interesting results. We believe that the best results in *SA* will be obtained by adopting such a hybrid approach.

In this paper, we present *iSEE*, a fully implemented sentiment extraction engine, which makes use of statistical methods, NLU techniques, common sense reasoning, and probabilistic inference to extract entity and feature specific sentiment from complex sentences and extended dialogs. Most of the components of *iSEE* are domain independent and the system can be generalized to new domains by simply adding domain relevant lexicons and without making changes to the processing engines. This makes *iSEE* easily extensible.

2 Brief review of related work

Much progress has been made in the field of sentiment analysis (Pang and Lee 08). Examples of applying the statistical approach include (Dave, Lawrence, and Pennock 2003), (Pang, Lee, and Vaithyanathan 2002), (Beineke et al. 2003), (Jin, Ho, and Srihari 2009). (Turney 2002) uses word-correlation algorithms for classifying entire documents based on their sentiment polarity. (Hatzivassiloglou and McKeown 1997) produced a list of seed words to determine whether a sentence contains positive or negative sentiments. 'SentiWordNet'(Esuli and Sebastiani 2006)is a lexical resource that aids opinion mining. Sentiment mining using lexicons is another widely followed approach (Yi and Niblack 2005), (Ding, Liu, and Yu 2008). Using appraisal groups of adjectives and modifiers to analyze sentiment is elaborated in (Whitelaw, Garg, and Argamon 2005). An interesting route followed by researchers is to perform text categorization to detect opinion clauses prior to sentiment analysis (Pang and Lee 2004), (Kim, Li, and Lee 2009), 'Opinion Finder' - (Wilson, Wiebe, and Hwa 2004). A hybrid system(Godbole, Srinivasaiah, and Skiena 2007) that combines lexical knowledge with statistical techniques to rank entities in a class based on public opinion in news and blogs has been proposed. Another approach to text analytics and sentiment analysis is based on NLU where a text undergoes syntactic, semantic and discourse analysis in order to extract "meaning" expressible in some formal notation (Yi et al. 2003).

Some approaches that do feature-specific sentiment extraction are 'Feature based summarization (FBS)' system developed by (Hu and Liu 2004) and the 'OPINE' system designed by (Popescu and Etzioni 2005). In the former, POS tag sequences are used to spot features which are further pruned to get frequent ones; opinions are extracted from the adjectives in the sentence. 'OPINE' is an information-extraction system that exploits syntactic details to produce feature specific sentiments and their relative quality. 'Opinion Observer' (Hu, Liu, and Cheng 2005) is another prototype system that analyzes and compares consumer opinions.

3 The Proposed Approach

We believe that the most effective role of statistical machine learning techniques is in the area of parsing. Consequently, *iSEE* uses a statistical parser. At the same time *iSEE* uses a rule-based linking engine to map syntactic constituents to semantic roles. *iSEE* also uses discourse analysis and other NLU techniques in conjunction with common sense reasoning, and probabilistic inference to extract entity and feature specific sentiments from complex sentences and extended dialogs. *iSEE* also makes use of several lexicons (general as well as domain specific) for enumerating entities, features and sentiment words. The architecture of iSEE is highly modular, and hence, easily extensible.

Functional Architecture

Fig.1 shows the functional architecture of the *iSEE* based Sentiment Analyzer. A description of each module follows.

Knowledge Repository (KR): The *KR* hosts knowledge consumed by various modules. This includes the *SMSese* dictionary, domain specific entity and feature lexicons, the set of linking rules that mediate the interaction between parses and semantic frames, *KRs* for common sense knowledge, idioms, and the sentiment vocabulary.

Pre-processor: The input to the system can be in various forms of unstructured text such as blogs, reviews, news articles, and call center logs. This input is filtered and cleansed by the pre-processor. The *SMSese* module replaces the SMSese words, shorthands and slangs known to the system and a domain specific spell checker corrects misspelled words in the sentence. In case of a blog, the pre-processor also captures the thread structure of the blog and its comments. The input is then split into a set of sentences by a sentence-splitter and passed on to the Syntactic Engine, the Semantic Engine and the Idiom module.

Syntactic Engine (SynE): The core of this engine is a statistical parser. Interfaces allow *iSEE* to work with any of the available parsers (See Results). The parser initially invokes a POS tagger to assign parts of speech to the tokens in the sentence. The Named Entity Extractor (*NEE*) in the Semantic Engine (*SynE*) identifies (and tags) domain relevant entities and features and conveys to the parser that these tagged named entities should be treated as a single token with the POS tag for a noun phrase (shown by the informer link from *NEE* to *SynE*). Similarly, the idiom module tags idioms in a sentence as a single token with an appropriate POS tag. The parser generates (typically, multiple) parses for such tagged sentences. It also classifies the sentence as a wh-question, an assertion, a comparison, confirmation seeking statement or a confirmation providing one.

Linking Engine (LE): The *LE* provides the critical linkage between the syntactic structure of a sentence and its

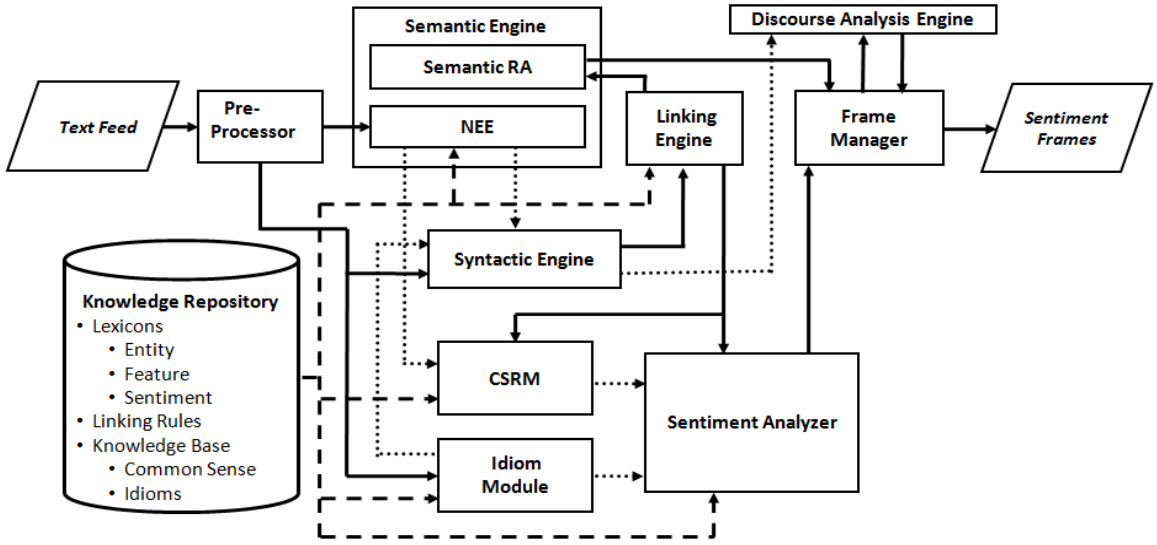

Figure 1: **Functional Architecture of 'iSEE' for Sentiment Analysis.** The bold lines indicate process flow, the dotted lines represent informer links and the dashed lines denote access to the knowledge repository.

meaning. It does so by identifying the mapping between the syntactic constituents of a sentence and the roles of the semantic frame that constitutes the meaning of the sentence. These mappings are expressed in terms of linking rules (LRs) whose antecedents are syntactic structures and whose consequents are semantic roles ((Fillmore, Johnson, and Petruck 2003), (Goldberg 1995)). The *LE* also has a rule prioritizer for resolving conflicts amongst overlapping rules. While an *LR* with a broader span typically overrides one with a narrower span, the *LRs* for conjunctive constructs override those for simple declarative sentences and those for comparative constructions have a higher priority over those for conjunctive and simple declarative constructs. The constituents extracted by the *LE* are passed to the Semantic Engine (SemE), the Common Sense Reasoning Module (CSRM) and the Sentiment Analyzer (SA). An example of an *LR* for a simple comparative construction - "⟨**Entity 1**⟩ *is* ⟨**better than**⟩ ⟨**Entity 2**⟩" is as follows:

```
<LR name="COMP_D" LRs_this_overrides = "D">
    <type-head type="VP" head="is,was,are,were">
        <type-head type="VBZ,VBP,VBD" head="is,was,are,were">
            <sibling side="right">
                <type-head type="JJR, RBR" head ="better,worse,
                superior" likely_semantic_role ="sentiment
                expression">
                    <sibling side="right">
                        <type-head type="IN" head = "than,to">
                            <sibling side="right">
                                <type-head type="NP, NN, NNS, NNP"
                                likely_semantic_role ="Entity2"/>
                                .......
        <sibling side="left">
            <type-head type="NP,NNP,NNS,NN"
            likely_semantic_role ="Entity1"/>
        </sibling>
    </type-head>
</LR >
```

In the XML specification for an *LR*, the *type-head* tag represents placeholders in terms of their POS tag (attribute 'type') and the actual word (attribute 'head'). The attribute, *likely_semantic_role*, is used to tell the semantic engine the potential roles each constituent might fill. When a sentence is tested against the *LR* described, the *LE* first detects a 'verb phrase (VP)' with head *is* in its parse. It checks if this *VP* contains a word with the POS tags for comparators *(JJR,RBR)* and if yes, marks this constituent as the likely *Sentiment Expression*. The phrase to the right of this word becomes the likely *Entity 2* and the one to the left of the *VP* is the likely *Entity 1*. As the current application focuses on sentiment analysis, the LRs encoded deal with high-frequency constructions used for expressing sentiments. There are a total of *25* such *LRs* that cover simple declarative constructs including conjunctions, comparative constructs, questions, and confirmation seeking and providing sentences.

Semantic Engine (SemE): The 'Named Entity Extractor (NEE)' of *SemE* plays a role early on in the processing of sentences. It tags entities and features of interest from preprocessed text through lexicalized lookup augmented with limited pattern matching. It also tags instances of entities about which common sense knowledge is available in the *KR*s. It alerts the POS tagger in the *SynE* to treat tagged named entities as a single token with a POS tag *NN* (for noun phrase). Depending on the degree of match between an n-gram(that has been tagged as a named entity) and a lexical entry in the *NEE*'s lexicon, the *NEE* assigns a probability to the POS tag *NN*. The sentence along with the the POS tags assigned by the *NEE* are presented to the POSTagger.

The *SemE* inspects interesting constituents extracted by the *LE* for the presence of annotated entities. It assigns semantic roles to them based on the mapping indicated by *LE*. In order to do so, it also checks that the 'potential filler' for a semantic role satisfies the requisite semantic type constraint (e.g., the entity should be a 'Service provider').

Idiom Module (IM): This component identifies and understands idioms. It forces the tokens in an idiom to be chun-

ked together with a POS tag *JJ* (adjective phrase). The idiom dictionary in *KR* supports *SA* when the sentiment expression is idiomatic in nature by providing the abstract meaning of an idiom. For example, the popular idiom *snail's pace* has the meaning, {*Domain = motion, Aspect = speed, Value = slow*}. The *Value* is utilized to infer the sentiment. For example, *snail's pace* in the context of *download speed* would signify a negative sentiment.

Common Sense Reasoning Module (CSRM): The *CSRM* adds the ability to do common sense reasoning to our engine. It uses common sense knowledge about limited aspects of the real- world (in KR) and an inbuilt inference engine. The rules of inference are triggered by certain words (often verbs) that head a class of metaphorical constructions. The properties/attributes of various entities that participate in the real-world situations relating to this word become antecedents for the inference rules. These rules are tiered according to their strength.

A special set of *LRs* , with the trigger words as placeholders, are fired on the parse of the potential sentiment expression extracted by the *LE* (When no *LR* is fired on a parse of a sentence, the special *LRs* are fired on the entire parse). They are further checked for class tags assigned by the *SemE* and knowledge about the tagged entities is retrieved from the *KR*. The inference engine generates inferences (to be consumed by the *SA*) on the basis of the evidence presented.

Say, the engine is processing the sentence: *TelAir downloads are like pouring honey from the jar*. The *LE* marks *pouring honey from the jar* as the sentiment expression. The special *LR* with *pour* as the placeholder identifies *honey* as the object being poured and *jar* as the source for pouring (semantic role assignments). The real world properties of honey and jar are fetched from the *KR*. A rule of inference for the word *pour* is that *"If the viscosity of the liquid being poured is high, the speed of the action is low"*. As the evidence gathered is that honey is a highly viscous liquid, the inference made is: {*Domain = motion, Aspect = speed, Value = low*}.

Sentiment Analyzer (SA): This component drills into the 'Sentiment Expression (SE)' extracted by the *LE* to detect and score the real sentiment. A *Sentiment* is typed *relative* or *absolute* depending on the sentence type (comparison or others) detected by the *SynE* and is quantified by its strength on a *0-5* scale and a +/- valence. In short: *Sentiment* = {*Type, Valence, Strength*}.

If idiom and/or *CSR* modules have generated inferences for an *SE*, the *value* of the inference is mapped to a sentiment grade using heuristics. The analytics engine in the *SA* assigns the corresponding strength and valence to the sentiment. For example, if the *SE* contains the idiom *snail's pace*, the inference produced by the *IM* is {*Domain = motion, Aspect = speed, Value = low*}. The *Value* 'low' in the domain of motion is mapped to the sentiment grade *negative* and the analytics engine assigns the corresponding score of *3* and valence -.

If no inferences are available, *LRs* specific to sentiment extraction called *SE-LRs* are fired on *SE* to extract potential sentiment words. If there are no extracted *SE*s, the *SE-LRs* are fired on the entire parse. Domain knowledge is used to generate inferences from these words and the *Value* of the inference is mapped to a sentiment grade. For example, the sentiment word *crawls* is inferred as {*Domain = motion, Aspect = speed, Value = low*} and the *Value* 'low' in the domain of motion is mapped to the sentiment grade 'negative'. If this doesn't yield a grade for the sentiment, a generic sentiment lexicon (which was build from a set of seed words expanded using 'WordNet') is used to grade the sentiment. The extracted words are also checked for being modifiers, comparators or negators using the respective lexicons in the *KR*. The analytics engine assesses the contributions of modifiers and negators to the sentiment extracted. While modifiers increase/decrease the sentiment strength, the negators often reverse the sentiment valence. For example, given the sentiment words *very good*, the modifier *very* increases the strength of the sentiment from *3* to *4*. The presence of comparators like *better* results in the sentiment being typed as a 'relative sentiment' and it's valence is set according to the polarity of the comparator. For example, for the sentence *TelAir is better than AirVoice*, the sentiment word *better* qualifies as a comparator with a + polarity. So, for the frame for the entity *TelAir*, the sentiment valence is + and for the *AirVoice* frame, the valence is -, with the type of both frames being 'relative'.

SA also uses domain knowledge about a feature to determine the sentiment value. It does so based on the utility of an associated feature. For example, cost has a negative utility while speed has a positive utility; at least in the context of broadband services. Hence, even though the sentiment word 'high' has a positive connotation in general, for the feature 'cost' it implies a negative sentiment.

Frame Manager (FM): The frame manager creates a *frame* for every unique entity and feature extracted by the *SemE*, wherein the semantic labels are mapped to the corresponding entity or feature. Different parses of a sentence may result in different constituents being extracted by the *LE* implying the creation of multiple frames. The *FM* removes duplicate frames, discards frames contained by other frames and sorts the remaining frames based on the spans over which the *LRs* that created them fired. Every unique frame is assigned a frequency score which is the ratio of the number of instances of this frame to the total number of frames generated over multiple parses of a sentence. If the frequency score of a frame is below a threshold value, it is eliminated.

The *FM* binds the extracted sentiment to the corresponding frames to create 'Sentiment Frames'. When no entity or feature is extracted by the *SemE* but a sentiment is extracted by the *SA*, a sentiment frame is still created with this sentiment. The Discourse Analysis Engine (*DAE*) acts on it at a later stage to fill in the missing entity/feature. The frame is also attributed a belief score depending on how the entity, feature and sentiment where extracted. (The belief score is higher for frames where the entity and/or feature were captured by *LRs*).

Discourse Analysis Engine (DAE): This module performs elementary frame level discourse analysis on the frames generated by *FM*. The Syntactic Engine informs it of the class of a sentence (a question, an assertion, a compari-

son or a confirmation seeking or providing statement) from which a frame was derived. For frames that were flagged for anaphora resolution, a trace along the previously occurring frames is done to identify an implicit feature/entity. The frames generated from confirmatory messages are often indicative of the sentiment for a question previously raised. The *DAE* steps in to accomplish this task.

An Illustrative Example

In this section, we take you through a pass of *iSEE* for sentiment analysis, with the help of an illustrative example. The outputs at various stages are as shown below:

- **Text Feed**
  TelAir's download speed is gr8 but its customer srvc is no better than AirVoice's.
- **Pre-processor**
  TelAir download speed is **great** but its customer **service** is no better than AirVoice's.
- **NEE (Semantic Engine) + Syntactic Engine**

```
(ROOT (S (S
    (NP(NP (POS TelAir's/Corp))(NN download speed/DS))
    (VP(VBZ is)(NP (JJ great)))
   )(CC but)
   (S
    (NP(PRP$ its)(NN customer service/CS))
    (VP(VBZ is)
     (ADJP (ADJP(RB no)(JJR better))
      (PP(IN than)(NP (POS AirVoice's/Corp)))
     )
    ))))
```

  Note that the named entities have appropriate class 'tags' and tokens within them are chunked together.

- **Linking Engine**
  - LR *D*: (NP (POS TelAir's/Corp)), (NN download speed/DS), (VP(VBZ is)(ADJP (JJ great)))
  - LR *COMPDPOS*:
    - (NP (PRP$ its)), (NN customer service/CS), (JJR better)
    - (NP (POS AirVoice's/Corp)), (NN customer service/CS), (JJR better)
- **Semantic RA (Semantic Engine**
  - Set 1: Provider = TelAir, Feature = download speed
  - Set 2:
    - Provider = it, Feature = customer service
    - Provider = AirVoice, Feature = customer service

  Note that the semantic roles 'Provider' and 'Feature' are assigned by correlating the class tags with linkages established by *LE*.
- **Idiom, Common Sense Reasoning Modules**
  No inferences for this example.
- **Sentiment Analyzer**
  - Sentiment 1: (JJ great) $\Rightarrow$ [Sentiment = {Grade : *very positive*, Type : *Absolute*, Valence : +, Strength : *5*}]
  - Sentiment 2: (RB no), (JJR better) $\Rightarrow$ [Sentiment = {Type : *Relative*, Valence : -, Strength : *3*}] [Sentiment = {Type : *Relative*, Valence : +, Strength : *3*}]

Sentiment Frame 1	
Statement Type	Assertive
Provider	TelAir
Feature	Download Speed
Frequency Score	0.31
Belief Score	0.95
Sentiment	
Sentiment Phrase/Word	great
Type	Absolute
Valence	+
Strength	5
Source of Information	Lexicon

Sentiment Frame 2	
Statement Type	Comparative
Provider	TelAir
Feature	Customer Service
Frequency Score	0.31
Belief Score	0.95
Sentiment	
Sentiment Phrase/Word	no better
Type	Relative
Valence	-
Strength	3
Source of Information	Lexicon

$<$

Sentiment Frame 3	
Statement Type	Comparative
Provider	AirVoice
Feature	Customer Service
Frequency Score	0.38
Belief Score	0.95
Sentiment	
Sentiment Phrase/Word	no better
Type	Relative
Valence	+
Strength	3
Source of Information	Lexicon

- **Discourse Analysis Engine + Frame Manager**
  For this example, the *DAE* resolves *it* in Frame 2 to *TelAir*.

The above example involved a relatively simple sentiment sentence. The *iSEE* system can handle extended discourse consisting of complex sentences referring to multiple products and features and involving comparisons, conjunctions, questions and the like. It can also deal with idiom and metaphor including productive ones in the limited domain of liquids using *CSRM*. For example, the system extracts the correct sentiments in the following dialog: S1: How is TelAir's download speed? S2: It is excellent. S1: But I warn you that TelAir's customer service is very bad. S2: Then tell me, how is AirVoice? S1: It is better than TelAir. S2: I was told that TelAir has a good customer service in my area. S1: How is its download speed? S2: It is better than AirVoice's. AirVoice downloads are like pouring molasses in winter. S1: I heard from my friends that AirWave has a good download speed.

4 Results

In order to benchmark our system and find room for improvement, we compared *iSEE* with the Sentiment Analyzer in the *FBS* system developed by (Hu and Liu 2004). *FBS* also performs feature specific sentiment extraction and assigns strength to these sentiments, which makes it a good standard to be compared against. The test data set for this experiment was a set of sentences annotated with feature or entity and the corresponding sentiment. They were gathered from product reviews in Amazon.com and Cnet.com.

In Table 1, we report the performance of *iSEE* and *FBS* on a text feed of 225 sentences from the DVD player reviews. The *SynE* used *Stanford NLP Parser* (NLP) for statistical parsing. The *LE* contained 25 different *LRs* covering commonly used constructions for expressing sentiment. The *SemE* was provided with a set of 4 entities and 15 features in the domain of DVD-Players.

Approach	Precision	Recall	F-Score
iSEE	0.93	0.85	0.89
FBS	0.94	0.65	0.77

Table 1: *iSEE, FBS - DVD players dataset*

It is apparent that *iSEE* has a much higher recall than *FBS* and it's precision is only marginally lesser than that of *FBS*. The higher F-score of 12% (absolute) achieved by *iSEE* over *FBS*, which is very encouraging.

To validate the argument that our approach to sentiment extraction is not specific to a domain, we tested *iSEE's* performance on a set of 242 sentences from the cellular phone reviews annotated by Liu and Hu. Table 2 shows the results.

Approach	Precision	Recall	F-Score
iSEE	0.94	0.84	0.88
FBS	0.95	0.65	0.77

Table 2: *iSEE, FBS - cellular phones dataset*

Even in this domain, the F-score of *iSEE* is more than 10% (absolute) higher than that of FBS.

In order to quantify the contribution of the linking rules and the common sense reasoning we tested *iSEE* without the *LE* and *CSRM*. The limited system had 8% lower precision and 13% lower recall (absolute %).

5 Conclusion

In this paper, we have described a flexible and extensible approach for solving the sentiment extraction problem. The resulting system, *iSEE*, makes use of statistical methods, classical NLU techniques and probabilistic inference to extract entity- and feature-specific sentiments from extended dialogs and complex sentences involving comparatives and conjunctions. The system makes use of common sense reasoning to understand productive metaphors within limited domains (currently, motion and liquids). The components of *iSEE* are domain independent and the system can be generalized to new domains by simply adding domain relevant lexicons. The system also analyzes idioms and metaphors for their sentiment. iSEE has its share of limitations. Irony, sarcasm, humour and other such subtle ways of sentiment expression are not handled by the current system. Currently, our focus is on identifying additional constructions for expressing sentiments and on expanding the common sense knowledge base to enable *iSEE* to draw inferences about a larger set of domains. The flexibility, modularity and extensibility of the engine provide an easy, fast and reliable method for developing solutions across different domains.

6 Acknowledgements

We would like to thank Amar Viswanathan, Bintu Vasudevan, Prasanna Venkatesh, Shabin Hashim, Suman Kumar Chalavadi, Swarna Latha Ramalingam, and Umadas Ravindran for their valuable contributions to the *iSEE* effort.

References

Beineke, P.; Hastie, T.; Manning, C.; and Vaithyanathan, S. 2003. An exploration of sentiment summarization. In *AAAI '03*.

Dave, K.; Lawrence, S.; and Pennock, D. M. 2003. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *WWW '03*.

Ding, X.; Liu, B.; and Yu, P. S. 2008. A holistic lexicon-based approach to opinion mining. In *WSDM '08*.

Esuli, A., and Sebastiani, F. 2006. Sentiwordnet: A publicly available lexical resource for opinion mining. In *LREC '06*.

Fillmore, C.; Johnson, C.; and Petruck, M. 2003. Background to framenet. *Int. Journ. of Lexicography*.

Godbole, N.; Srinivasaiah, M.; and Skiena, S. 2007. Large-scale sentiment analysis for news and blogs. In *ICWSM*.

Goldberg, A. 1995. Constructions: a construction grammar approach to argument structure. *U. of Chicago Press*.

Hatzivassiloglou, V., and McKeown, K. R. 1997. Predicting the semantic orientation of adjectives. In *Proc. 8th conference on European chapter of the ACL*.

Hu, M., and Liu, B. 2004. Mining and summarizing customer reviews. In *SIGKDD '04*.

Hu, M.; Liu, B.; and Cheng, J. 2005. Opinion observer: analyzing and comparing opinions on the web. In *WWW*.

Jin, W.; Ho, H. H.; and Srihari, R. K. 2009. Opinionminer: a novel machine learning system for web opinion mining and extraction. In *SIGKDD '09*.

Kim, J.; Li, J.-J.; and Lee, J.-H. 2009. Discovering the discriminative views: Measuring termweights for sentiment analysis. In *IJCNLP '09*.

NLP, S. http://nlp.stanford.edu/software/lex-parser.shtml.

Pang, B., and Lee, L. 2004. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL '04*.

Pang, B.; Lee, L.; and Vaithyanathan, S. 2002. Thumbs up? sentiment classication using machine learning techniques. In *EMNLP '02*.

Popescu, A.-M., and Etzioni, O. 2005. Extracting product features and opinions from reviews. In *EMNLP '05*.

Turney, P. D. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *ACL '02*.

Whitelaw, C.; Garg, N.; and Argamon, S. 2005. Using appraisal groups for sentiment analysis. In *CIKM '05*.

Wilson, T.; Wiebe, J.; and Hwa, R. 2004. Just how mad are you? finding strong and weak opinion clauses. In *NCAI*.

Yi, J., and Niblack, W. 2005. Sentiment mining in webfountain. In *ICDE '05*.

Yi, J.; Nasukawa, T.; Bunescu, R.; and Niblack, W. 2003. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *ICDM '03*.