CornPittMich Sentiment Slot-Filling System at TAC 2013

Carmen Banea*, Yoonjung Choi, Lingjia Deng**, Ozan Irsoy***, Detian Shi***,
Claire Cardie***, Rada Mihalcea* and Janyce Wiebe****

* University of Michigan
** University of Pittsburgh
***Cornell University

Abstract

We describe the 2013 system of the CornPittMich team for the KBP English Sentiment Slot-Filling (SSF) task. The central components of the architecture are two fine-grained opinion analysis systems. For each query, we select the top N documents based on a document-level relevance measure, and process each sentence therein to identify expressions of sentiment along with their source and target. The KBP knowledge base is used for the efficient retrieval of relevant documents.

1 Introduction

This paper describes a collaboration between Cornell University, the University of Pittsburgh and the University of Michigan to develop a system for the English Sentiment Slot-Filling (SSF) task as part of TAC 2013. Our goal was to combine two existing systems for the fine-grained analysis of opinionated text — OpinionFinder (Wilson et al., 2005a) and the CRF- and ILP-based opinion analysis system of Yang and Cardie (2013). As newbies to the Knowledge Base Population (KBP) tasks, however, we misinterpreted the initial instructions — we understood that only one document per query would require processing — and allotted not nearly enough time to construct what would have to be a much more complex system than planned. The good news is that we succeeded in creating the joint system and submitted results. Sadly, the results submitted were the very first set of results that the system produced on either the training or the test data.

Below we first describe the architecture of our system (Section 2) and each of its components. We then present our results along with an initial analysis of system errors (Section 3).

2 System Architecture

The high-level system architecture of the CornPittMich (CPM) system is shown in Figure 1. Given a query, the system first retrieves all documents from the KBP corpus that mention the query entity using either its full name or any known alternatives. This document set is further filtered with respect to named entities (NEs) from the provided BBN SERIF annotations, leaving only documents that contained at least one NE in addition to the query entity. Then we apply the opinion analysis components to the top N remaining documents to identify potential slot-fillers (i.e., the targets or opinion holders) associated with the opinion query entity. These are further filtered in a post-processing phase.

In the next section, we provide a short description of each component of the overall system.

2.1 Preprocessing: Resolving Named Entities with the Knowledge Base

To optimize query-time access to the large corpus released as part of the KBP SSF task, we sought to resolve each SERIF named entity (and SERIF-identified coreferent mentions) to its unique knowledge base entry.

More specifically, a query in the KBP SSF task is composed of a named entity (i.e., the query entity) and a sentiment relation (positive/negative-from or positive/negative-toward). The answer to the query

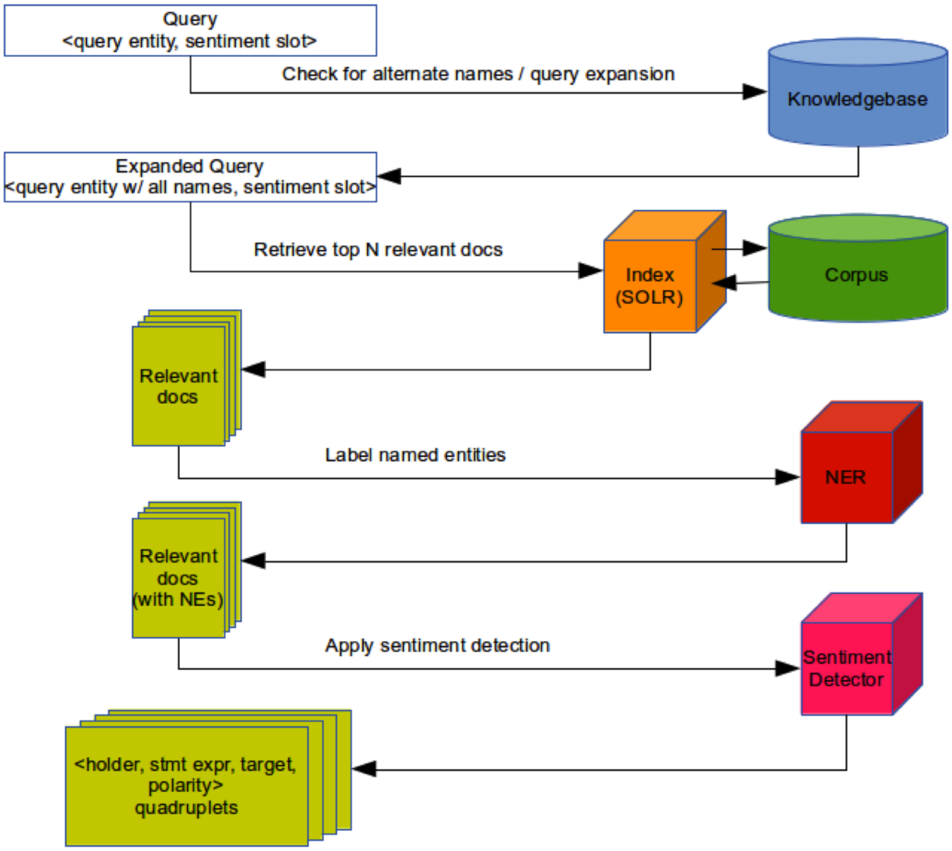
Query
<query entity, sentiment slot>
Check for alternate names / query expansion
Knowledgebase
Expanded Query
<query entity w/ all names, sentiment slot>
Retrieve top N relevant docs
Index
(SOLR)
Corpus
Relevant
docs
Label named entities
NER
Relevant
docs
(with NEs)
Apply sentiment detection
Sentiment
Detector
<holder, stmt expr, target,
polarity>
quadruplets

Figure 1: System Architecture

should consist of a named entity that occurs in the prescribed sentiment relation with respect to the query entity. In the majority of queries, the named entity is also accompanied by a knowledge base (KB) identifier that uniquely denotes the entity. During preprocessing, we seek to resolve all NEs and coreferent mentions to their entry in the KB. With proper indexing, this allows us to retrieve at evaluation time only those documents that contain information pertaining to the query entity.

Difficulties stem from the non-canonical representation of the entities in natural text, which is largely caused by their orthographic realization or alternative representations. In order to alleviate this problem, we mine alternative mention representations from Wikipedia article annotations, and implement a voting mechanism that allows the most likely resolutions to surface.

In particular, we first mine the entities' surface forms using Wikipedia as a corpus. This is accomplished by leveraging the Wikipedia editors' annotations, which resolve mentions to Wikipedia articles by employing a framework of internal hyperlinks. Based on these metrics, the second step consists of computing the likelihood that a mention would resolve to a particular knowledge base entry. Finally, the last step involves processing the actual SERIF annotations and providing a list of potential knowledge base resolutions for each named entity. We explain the latter in more detail below.

Given a document, the SERIF-formatted annotations provide a set of mentions for each named entity. For example, for a named entity we may encounter the following mentions: "Barack Obama," "he," "the president," "B. Obama," "Barack Hussein Obama," etc. Each of these mentions allows the referred entity to gain a stronger contour, thus allowing these non-canonical representations, to cast their vote for the most likely knowledge base resolutions, by employing the Wikipedia resolution likelihoods computed earlier. At the end, for each named entity that we were able to resolve, a list of possible knowledge base resolutions is provided, with candidates ranked from the strongest to the weakest.

2.2 Indexing, Parsing and Document Retrieval

We used Apache's Lucene-based Solr search platform to index both the corpora and the KBP knowledge base itself. The corpora are searchable by keyword and document ID as well as via any metadata provided along with the raw text (such as `author`). The latter is helpful for retrieval of discussion form posts for which the query entity is the author.

Discussion forum posts are separated and each treated as a single document. Unique IDs for these were created by adding a post ID as a prefix to the original document ID. This is done because each post is potentially authored by a different person; thus, they are usually disjoint.

Additional preprocessing of the documents was done to save time during testing. We use Solr to retrieve the top 2000 documents[1] for each test query, half based on author metatdata information and half based on keyword search. The retrieved documents are parsed using the Stanford Parser augmented with the original byte offsets for each token so that subsequent tokenization transformations can be inverted.[2]

Ultimately, for each retrieved document, a search returned raw text, structured text, parse trees for sentences, and sentences as ordered lists of tokens. In addition, some sanitization was done to strip out HTML artifacts such as angle brackets and other invalid characters. Furthermore, in order to produce text that would be easy to process by the sentiment analysis systems, metadata blocks like author, sender information, and other sections irrelevent to sentiment analysis were stripped out before handing over to sentiment systems for analysis.

2.3 Sentiment Analysis

The documents retrieved for each query are processed by one, or both, of the sentiment analysis systems — OpFind+Heur and CRF+ILP. The goal of these systems is to identify opinion expression tuples from each document: [holder, expression, target, polarity]. These will be filtered in a postprocessing step that removes duplicates and retains only those slot-fillers that match the query specifications.

2.3.1 Sentiment Detection: OpinionFinder and Heuristics

The OpFind+Heur system used the OpinionFinder system (Akkaya et al., 2011) to identify sub-

[1] This was due to parse tree creation times.

[2] We spent a lot of time dealing with byte offset misalignment errors and likely did not manage to fix all of them.

jective sentences and to detect sentiment expressions. We also use two widely-used subjectivity lexicons (Wilson et al., 2005b; Stone et al., 1966) to count the number of positive and negative words in each sentence. Additionally, we employ the idea of "good-for/bad-for" concepts (Deng et al., 2013) to identify actions that can cause or produce sentiment. For example, "helping" is good-for the entity that is helped; while something like "kicking" is presumably bad-for the entity that is kicked. Thus, we assume that the sentiment of the actor toward the object of a good-for verb is positive and the sentiment of the actor toward the object of a bad-for verb is negative. We manually identified good-for and bad-for verbs fram FrameNet. Based on all of the above, the majority vote of the positive and negative words determines the sentiment of all subjective expressions in the sentence.

To associate an opinion holder and a target for the identified sentence-level sentiment value, we combine a heuristic method and a machine learning-based method. The heuristics are obtained from the training data in which 65.8% of annotations (645/980) are from discussion forums; among these, in 81.2% (524/645), the opinion holder is the author. The most frequent grammatical roles for a holder and a target are the subject and object (27.04 percent of holders are *subj* and 8.96 percent of holders are *obj*; 21.79 percent of targets are *subj* and 9.23 percent of targets are *obj*). Thus, as the heuristic method, we apply simple rules such as:

- If a document has author information, the opinion holder is the author and the target is a *subj* or *obj* in the sentence.
- If not, the opinion holder is in the *subj* position in the sentence and the target, an *obj* position.

More specifically, we assume that the *subj* (or *obj*) is the nearest *subj* (or *obj*) to the sentiment expression in the parse tree.

These rules, however, cannot cover all cases. So, we also create two classifiers — one for opinion holder detection and one for target detection. The candidates are all named entities, and we extract several features such as an opinion word, POS of an opinion word, POS of a candidate, distance between a candidate and an opinion word in the parse tree, the shortest distance from a candidate to a term from the *subj* in the parse tree, the shortest distance from a candidate to a term in the *obj* in parse tree, has author information or not, named entity type, overlapping parsing information between a candidate and an opinion word, (only for holder detection) is a candidate pronoun or not, and (only for holder detection) is a candidate an author or not. With these festures, we apply the J48 Decision Tree algorithm from Weka. Models are trained over the training data. We conduct 10-fold cross validation to test trained classifiers with the training data. In the target model, the precision is 0.582, the recall is 0.58, and the f-measure is 0.563; in the holder model, the precision and recall is 0.864 and the f-measure is 0.863.

2.3.2 CRFs to Extract Opinion Relations

The CRF+ILP system uses a Conditional Random Field (CRF) (Lafferty et al., 2001) and Integer Linear Program (ILP) based opinion extraction system (Yang and Cardie, 2013) for within sentence identification of subjective expressions, opinion targets and (possibly implicit) opinion holders. The system is trained over the MPQA corpus (Wiebe et al., 2005) and models a sentence as a sequence of segments, by relaxing the Markov assumption of classical CRFs, in turn, allowing the incorporation of segment-level labels (Yang and Cardie, 2012). For the KBP SSF task, we only identify *Direct Subjective Expressions*, e.g. "criticized", "like", "pit X against Y". Integer Linear Programming is used to coordinate the construction of opinion *relations* from the set of possible subjective expresssions, targets and holders. This component establishes the connections between expression-holder and expression-target annotations.

Since there is no sentiment value (i.e., polarity) extracted, the model is not currently a standalone system for extraction of subjective expressions with positive or negative sentiment. We rely on OPFIND+HEUR to assign the proper polarity.

2.4 Postprocessing

We detect duplicate opinions extracted by the CRF+ILP and OPFIND+HEUR systems and return only one of them. First, the polarities of any pair of duplicate opinions must be the same. Then, if the offsets of the holders, the targets and the opinion expressions overlap, we assume that the two opinions are the same and report only one of them. If the two opinions are from difference sentences but both the holders and the targets refer to the same entity according to the named entity coreference annotations, then we report the two opinions in one line but with their offsets delimited by ",", as required in the task description.

After duplicate detection, we select the opinions in accordance with the query type. For example, if the query type is "pos-from", then we filter out all the negative opinions and consider only the opinions whose targets are the query entity, according to the named entity coreference information. Specifically, if the query entity has an id in the knowledge base, we will select the opinions that have targets with the same id. We report any opinion returned after matching the query or "NIL" if there is no opinion.

3 Results and Analysis

Results for our system were:

Recall: (7+0) / (904+0) = 0.008

Precision: (7+0) / 70 = 0.1

F1: 0.014

So what happened? In the end, due to time constraints, we processed only the top 10 retrieved documents for each query: OpFind+Heur processed these documents and relied on CRF+ILP only for documents for which no slot-filler could be identified. The submitted results were furthermore the very first results obtained on either the training or test data. Additional component-wise analysis is provided below.

3.1 Document Retrieval

The total number of retrieved "documents" for our system is actually 1529 (not 10) since we consider each post in a discussion forum to be its own document. Each of these is processed separately.

Based on the gold standard answer key for the test queries, there are 716 documents that contain the desired slot-fillers. Among our 1529 retrieved documents, only 89 have a correct slot-filler. Thus, our document retrieval precision is 5.82% (89/1529) and recall is 12.43% (89/716).

3.2 Slot Filler Detection

Among 89 correctly retrieved documents, we extracted some filler information (potentially incorrect) from only 13 documents. That is, our system extracted no information from 76 documents. Among the 13 documents, we only provide seven correct answers (and six incorrect answers). In three of the incorrect cases, the sentiment phrase and sentence (i.e., relation justification) is correct, but the extracted filler is wrong.

3.3 OpFind+Heur

OpFind+Heur produces 1439 responses of which 26 are correct. The precision is 0.018, recall is 0.029, and F-measure is 0.022. Even though it extracted more correct responses, many incorrect responses were also detected. Furthermore, OpFind+Heur detected sentiment information from 38 documents among the 89 correctly retrieved documents. That is, it missed 51 documents.

When considering only correctly retrieved documents, the number of slot-fillers in the gold standard 183; OpFind+Heur finds 119 of which 26 are correct (precision: 0.218, recall: 0.142, F-measure: 0.172).

We observe that in many of these cases, the opinion holder is an author, which indicates the importance of successfully extracting author information and tying it to the opinion (potentially relying on coreference resolution).

3.4 CRF+ILP

CRF+ILP processed a subset of the queries (33 queries in total) — only those for which OpFind+Heur failed to produce a response (i.e., returned NIL). This was mainly due to time constraints. For each such query, CRF+ILP also processed only the top 10 documents in the relevant set.

Of the 33 queries, CRF+ILP finds relevant information from only three, producing five responses in total. None of the responses was fully correct. In one response, the filler was *inexact*, including irrelevant text around the correct entity name ("Spotify a lot" instead of "Spotify"). In another response, the system captures the correct target as a pronoun, however it is not connected to an entity due to lack of coreference resolution ("it" instead of "Tumblr").

The other cases involved more complex expressions for which the system is unable to identify the actual subjective expression bearing sentiment.

4 Conclusions

Note to selves: start earlier; listen to advisors who emphasized the importance of constructing and evaluating a simple end-to-end system to start; make use of even the limited training data to re-train components; start earlier; find and fix byte offset problems early and often; listen to advisors.

More seriously, important goals for us for next year are will be to deal more effectively with the conversational text of the discussion forums, to scale our methods so that more documents can be processed, and to investigate the reasons for the especially low recall.

References

Cem Akkaya, Janyce Wiebe, Alexander Conrad, and Rada Mihalcea. 2011. Improving the impact of subjectivity word sense disambiguation on contextual opinion analysis. In *Proceedings of the Fiftennth Conference on Computational Natural Language Learning (CoNLL-2011)*, pages 87–96. Association for Computational Linguistics.

Lingjia Deng, Yoonjung Choi, and Janyce Wiebe. 2013. Benefactive and malefactive event and writer attitude annotation. In *51st Annual Meeting of the Association for Computational Linguistics (ACL-2013, short paper)*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

P.J. Stone, D.C. Dunphy, M.S. Smith, and D.M. Ogilvie. 1966. *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press, Cambridge.

J. Wiebe, T. Wilson, and C. Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2):165–210.

Theresa Wilson, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005a. Opinionfinder: A system for subjectivity analysis. In *Proceedings of HLT/EMNLP 2005 Interactive Demonstrations*, pages 34–35. Association for Computational Linguistics.

Theresa Wilson, Janyce Wiebe, , and Paul Hoffmann. 2005b. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLP/EMNLP*, pages 347–354.

Bishan Yang and Claire Cardie. 2012. Extracting opinion expressions with semi-markov conditional random fields. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1335–1345, Jeju Island, Korea, July. Association for Computational Linguistics.

Bishan Yang and Claire Cardie. 2013. Joint inference for fine-grained opinion extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1640–1649. Association for Computational Linguistics.