Bootstrapping Polar-Opposite Emotion Dimensions from Online Reviews

Luwen Huangfu and Mihai Surdeanu
University of Arizona
Tucson, AZ 85719, USA
{huangfuluwen, msurdeanu}@email.arizona.edu

Abstract

We propose a novel bootstrapping approach for the acquisition of lexicons from unannotated, informal online texts (in our case, Yelp reviews) for polar-opposite emotion dimension values from the Ortony/Clore/Collins model of emotions (e.g., *desirable/undesirable*). Our approach mitigates the intrinsic problem of limited supervision in bootstrapping with an effective strategy that softly labels unlabeled terms, which are then used to better estimate the quality of extraction patterns. Further, we propose multiple solutions to control for semantic drift by taking advantage of the polarity of the categories to be learned (e.g., *praiseworthy* vs. *blameworthy*). Experimental results demonstrate that our algorithm achieves considerably better performance than several baselines.

Keywords: bootstrapping, semantic drift, limited supervision

1. Introduction

Bootstrapping is a lightly-supervised learning approach in which supervision comes in the form of a small number of initial examples (or seeds). While bootstrapping is an attractive choice for NLP, the limited supervision involved also yields important drawbacks. First, the training of bootstrapping models often "drifts" semantically from the original task into different tasks (e.g., from learning women names into learning flower names). Second, the lack of labeled data (i.e., only a small set of seed examples is annotated) impedes the capacity of the model to correctly assess the quality of the generated model during training.

In this paper we propose solutions for the above issues in the context of learning lexicons for the emotion dimensions (e.g., Desirability, Praise-/Blame-worthiness) necessary to assemble the Ortony/Clore/Collins (OCC) cognitive model of emotions (1990). According to the OCC framework, 22 different *emotion types* are generated from several *emotion dimensions*. In this work, we focus on the dimensions of Desirability, Praise-/Blame-worthiness and Likelihood because they are central emotion dimensions that contain both positive values (e.g., *desirable, praiseworthy, certain*) and negative values (e.g., *undesirable, blameworthy, likely*). For example, the emotion type *JOY* combines the dimensions Desirability with value *desirable* and Likelihood with value *certain*.

The contributions of this work are:

1. We propose multiple lightly-supervised solutions for the acquisition of emotion dimensions that control for semantic drift by taking advantage of the polarity of the classes to be learned (i.e., positive/negative appraisals).

2. We introduce an effective strategy to softly label unlabeled terms, i.e., unlabeled terms are assigned a value that indicates how close they are to a given category, and use these soft labels to better estimate the quality of extraction patterns in the above bootstrapping approaches.

3. We show that multiple resources (WordNet, word embeddings that project words in a continuous vector space that capture distributional similarity, and edit distance similarity) all help for the above two contributions, and are complementary to each other.

4. We empirically demonstrate that our approach outperforms several strong baselines for the acquisition of emotion dimensions lexicons from informal texts such as product reviews on the web.

2. Related Work

In the vast bootstrapping literature, a few works attempted to address the two limitations mentioned in the introduction. With respect to mitigating semantic drift, Kozareva and Hovy (2010) used stronger constraints for their lexicon extraction patterns, encouraging them to stay within the desired category to be acquired. Yangarber (2003) proposed "counter training", which introduces competition between the multiple categories (e.g., lexicon or event types) that are learned simultaneously (i.e., they are not allowed to overlap). This idea was generalized by the NELL system (Carlson et al., 2010). McIntosh and Curran (2010) extended counter training with negative categories that are discovered on the fly. Our approach is closest to counter training, with the extension that we propose multiple "soft" exclusion criteria.

With respect to the better handling of unlabeled data, Gupta and Manning (2014) improved the scoring of extraction patterns by predicting the labels of unlabeled terms, and using this information to better estimate the precision of the candidate patterns. Gupta and Manning (2015) extended this idea with a k nearest neighbors (kNN) formulation that expands the labeled training data with unlabeled entities that are close (according to kNN) to seed examples. Popescu and Etzioni (2005) applied a similar idea to the extraction of opinion words, where the unlabeled terms are labeled using a combination of syntactic, WordNet constraints, and relaxation labeling. Our work builds on these ideas with a simpler approach (no classifier is used). We also investigate more resources to measure the distance

Algorithm 1: Bootstrapping for emotion dimensions

input : A set of documents $\mathcal{D}$; seed words $\mathcal{S}$ for k emotion dimension values, $k \in \{$*praiseworthy*, *blameworthy*, *desirable*, *undesirable*$\}$

1 $\mathcal{Z} = \langle\rangle$ // stores extraction patterns for each k
2 $\mathcal{E} = \mathcal{S}$ // stores terms for each k
3 **foreach** epoch t **do**
  // expand the known set of terms:
4 **foreach** dimension value k **do**
    // we denote with $-k$ the dimension value opposite to k, e.g., if k is *desirable*, $-k$ is *undesirable*; if k is *praiseworthy*, $-k$ is *blameworthy*
5    $E(k) = \text{expandTerms}(\mathcal{E}(k), \mathcal{E}(-k), \mathcal{D})$
  // discover new extraction patterns:
6 **foreach** dimension k **do**
7    $P(k) = \text{extractAndRankPatterns}(E(k), E(-k))$
    // keep most relevant patterns:
8    $\mathcal{Z}(k) = \mathcal{Z}(k) + \text{getTop}(P(k))$
  // discover new terms:
9 **foreach** dimension k **do**
10   $T(k) = \text{extractAndRankTerms}(\mathcal{Z}(k), \mathcal{Z}(-k))$
    // keep most relevant terms:
11   $\mathcal{E}(k) = \mathcal{E}(k) + \text{getTopWithPolarityChecking}(T(k))$

output: $\mathcal{E}$

from known examples, ranging from WordNet (Miller et al., 1990) to `word2vec` (Mikolov et al., 2013), and show that they provide complementary information.

3. Approach

Algorithm 1 lists our proposed algorithm that extracts lexicons corresponding to emotion dimension values (or categories). Our algorithm builds on the traditional bootstrapping approach, which starts with a small set of seed examples, and alternates between learning extraction patterns and using them to discover new information (Riloff, 1996; McIntosh and Curran, 2008).

There are two fundamental differences between our approach and previous work. First, by using external information such as word embedding similarity, we expand the current set of acquired terms for each category, which are then *softly* labeled with category information (lines 4 – 5). The expanded term set is then used for the discovery of new extraction patterns (lines 6 – 8). Second, unlike McIntosh and Curran (2008)'s approach, which defined mutually exclusive categories, we remove the hard mutual-exclusivity constraint between categories. Instead, by taking advantage of the inherent polarity of the emotion dimension values, e.g., *desirable* is the opposite of *undesirable*, we introduce multiple soft constraints between opposite categories (k vs. $-k$). Such constraints are used throughout the algorithm (lines 5, 7, 10, 11). We detail these steps next.

3.1. Term Expansion

Expanding the terms (discussed below in this section) used for pattern ranking (see Section 3.2.) is important as it mitigates the sparsity of pattern-based methods with other complementary resources (akin to co-training (Blum and Mitchell, 1998)). We show in Section 4. that this step is important, especially in domain-specific settings, which are driven by "little data" scenarios.

The algorithm for term expansion relies on three resources: edit distance, word embeddings, and WordNet. We use these to generate candidate terms as follows: (a) We calculate the edit distance between every term in our corpus and the current term pool for category k ($\mathcal{E}(k)$); we consider new terms as candidates for category k if their edit distance is below a threshold (we used the same formula and threshold as (Gupta and Manning, 2014)). (b) We compute the average cosine similarity between every term's word embedding vector and the vectors of known terms in category k, and consider a term as a candidate if this similarity is above a threshold.[1] (c) Lastly, we use WordNet synonyms, derived wordforms, direct hypernyms and hyponyms of the term currently in the pool of category k as expansion candidates for k, and their antonyms as candidates for $-k$.

It is important to note that these three resources have complementary strengths and weaknesses. For example, edit distance naturally captures misspellings, but it also introduces false positives, e.g., "goods" as a candidate for the dimension containing "good". However, using the similarity of word embedding vectors mitigates this problem because the distributional similarity of the two words is low. On the other hand, antonyms (e.g., "good" vs. "bad") tend to have a high distributional similarity (Yih et al., 2012). WordNet addresses the latter problem, but its coverage is far more limited than that of word embedding models, which limits its applicability to domain-specific texts.

In order to allow these different resources to help each other, they have to interact. To do this, we assign to each candidate term produced above a score that combines the three resources:

$$\text{cscore}(c_k) = EDP - EDN + EMP - EMN + WNP - WNN \quad (1)$$

where c_k is a candidate term for category k. EDP and EDN use the discretized **e**dit **d**istance from the current **p**ositive/**n**egative entities (i.e., terms belonging to category k vs. $-k$) of Gupta and Manning (2014). EMP and EMN are the average cosine similarities between the **em**bedding vector of c_k and the embedding vectors of the terms in the **p**ositive category k (for EMP) and **n**egative category $-k$ (for EMN).

Lastly, WNP and WNN measure the overlap of the candidate term WordNet synset information with the terms in the **p**ositive and **n**egative category, respectively, as follows. In particular, the WNP score of a candidate term c_k is computed as the term overlap between the synonyms of c_k (Syn(c_k)) and the set of expanded terms in k from the previous epoch $E(k)^{t-1}$ plus the the term overlap between the antonyms of c_k (Ant(c_k)) and $E(-k)^{t-1}$.[2] This count is then normalized. WNN is computed similarly, but with the two categories (k and $-k$) flipped:

$$\text{WNP}(c_k) = \left(\frac{n_1}{|\text{Syn}(c_k)|} + \frac{n_4}{|\text{Ant}(c_k)|}\right) \times log\left(\sum_{l=1}^{4} n_l\right) \quad (2)$$

[1] We used word embeddings of 200 dimensions generated with `word2vec`'s skip-gram algorithm over the English Gigaword. We used 0.6 for the threshold. These values were not tuned.

[2] We used 1st and 2nd order synonyms and antonyms. For example, for each c_k, we construct Syn(c_k) from its synonyms and the synonyms of the synonyms. $E(k)^0$ is set to the seed terms of k.

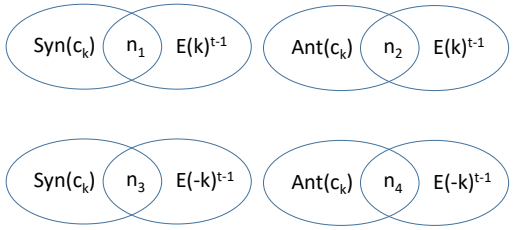

Figure 1: Illustration for the computation of the WNP and WNN scores for a term c_k, using the overlap between its synonyms (Syn(c_k)) and antonyms (Ant(c_k) vs. the category at hand (k) and its polar opposite ($-k$).

$$\text{WNN}(c_k) = (\frac{n_3}{|\text{Syn}(c_k)|} + \frac{n_2}{|\text{Ant}(c_k)|}) \times log(\sum_{l=1}^{4} n_l) \quad (3)$$

where n_1, n_2, n_3, and n_4 are the number of terms in the set intersections shown in Figure 1. The log component follows the intuition of (Riloff, 1996) to promote terms that are frequent; but here we adapt it to use the size of the overlap between WordNet synsets and the pools of known terms.

We use the score in Eq. 1 in multiple ways. First, we implement two "cautiousness" (Collins and Singer, 1999) constraints, i.e., we accept only candidate terms that: (a) have all the following three conditions true: EDP$\geq$ EDN, WNP$\geq$ WNN, EMP$\geq$ EMN (i.e., their association with the positive class is stronger than the one with the negative class under all resources); and (b) have at least one of the constraints satisfied: EDP$-$EDN $\equiv$ 1.0[3], WNP $-$ WNN $\geq \lambda_2$, EMP $-$ EMN$\geq \lambda_3$. Second, we use the score to estimate the quality of extraction patterns, as detailed next.

3.2. Pattern Learning

In this work, we define patterns similarly to McIntosh and Curran (2008), as five- or four-grams over surface tokens that include the term under consideration.

Our pattern learning algorithm expands upon existing methods by taking advantage of: (a) the set of expanded terms E, and (b) the polarity of the emotion dimensions to be learned. For each category k, each pattern (or template) t is assigned the following relevance weight:

$$\begin{gathered}\text{weight}(t \in \mathcal{Z}(k)) = \\ (\textstyle\sum_{c \in E(k)} \text{score}(t,c) - \sum_{c \in E(-k)} \text{score}(t,c)) \\ \times log(\textstyle\sum_c \text{frequency}(t,c))\end{gathered} \quad (4)$$

where frequency(t,c) captures the number of times a pattern t and a term c matched in the text; and score(t,c) measures the association between the pattern and the term c using the formula:

$$\text{score}(t,c) = log(\text{cscore}(c)\frac{p(t,c)^2}{p(t)p(c)} + 1) \quad (5)$$

Eq. 5 builds upon the squared mutual information (MI^2) formula of McIntosh and Curran (2008). We follow McIntosh and Curran (2008) by choosing squared MI over plain MI.[4] We weigh the MI^2 term by the association strength between term c and the current category k (i.e., cscore from Eq. 1). Further, we replace the hard mutual exclusive constraint from McIntosh and Curran (2008), which does not allow term and patterns to belong to multiple categories, with the soft constraint captured in Eq. 4, where opposite categories (k and $-k$) compete for pattern t.

Patterns are ranked in descending order of their weight, and the top M patterns are added to the cumulative pattern pool of category k in each epoch.

3.3. Term Learning

Lastly, we add new terms to the pool of known terms ($\mathcal{E}$) using the patterns previously learned. Terms are ranked by the following formula:

$$\begin{gathered}\text{weight}(c \in \mathcal{E}(k)) = \\ (\textstyle\sum_{t \in Z(k)} \text{score}(c,t) - \sum_{t \in Z(-k)} \text{score}(c,t)) \\ \times log(\textstyle\sum_t \text{frequency}(c,t))\end{gathered} \quad (6)$$

Eqs. 4 and 6 are nearly symmetrical for terms and patterns. Similarly, score(c,t) measures the association between the pattern and a term c using the formula:

$$\text{score}(c,t) = log(\text{tscore}(t)\frac{p(t,c)^2}{p(t)p(c)} + 1) \quad (7)$$

Eqs. 5 and 7 are symmetrical for terms and patterns, but in Eq. 7 tscore(t) is set to 1.0 because we rely solely on patterns from the pools of known patterns (i.e., no pattern expansion was implemented).

Similar to the previous step, the candidate terms are ranked in descending order of their weight, and the top N are added to the cumulative pool of terms.[5] Additionally, this step implements the "cautiousness" constraint from Section 3.1..

4. Experiments

Corpus: we are interested in learning from small datasets containing informal language. Here, we used a corpus of reviews for 1,439 sellers of e-cigarettes that we collected from Yelp. We downloaded all reviews from each page. The final corpus contains 1,600,151 tokens. The texts were stripped of URLs, then tokenized using CoreNLP (Manning et al., 2014). Note that our dataset contains only free text, without any hashtags or emoticons. We split data into training (75%) and testing (25%). We used the training dataset solely to tune the model's hyper parameters: λ_1, λ_2, and λ_3 from Section 3., as described later in this section. The tuning happened exclusively on the training dataset.

Terms and patterns: we consider terms to be single-word adjectives, nouns, verbs, or adverbs (e.g., "horrific"), and patterns to be 4- or 5-grams surrounding them (e.g., "avoid the _ burn lead" pattern captures the previous term).

[3]Note that the values of EDP and EDN are discretized edit distances, and can take only values of 0 or 1 (Gupta and Manning, 2014).

[4]Similar to (McIntosh and Curran, 2008), we observed that this formula performed better than $p(t|c)$ and regular MI.

[5]To mitigate the sensitivity to low frequencies, we set a candidate term's weight to 0 if $\sum_t \text{frequency}(c,t) \leq 3$ in Eq. 6. These terms are separately ranked using Eq. 1 instead.

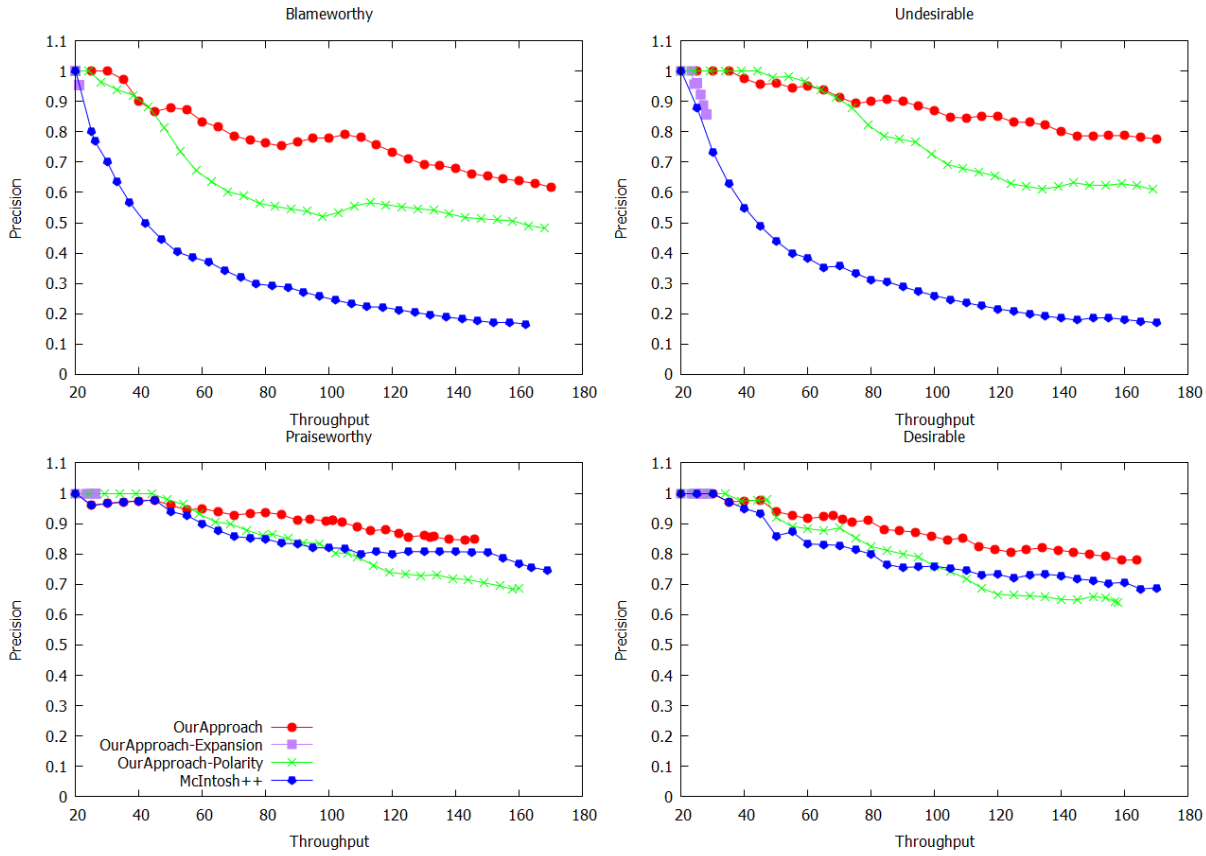

Figure 2: Precision/throughput curves of our approach and the three baselines. All systems were run for up to 30 epochs.

Evaluation measures: we report cumulative precision and throughput of terms for 30 epochs. Here, cumulative precision means the fraction of the words retrieved that are correct for the given category; cumulative throughput means the number of words that are retrieved by a system. All extracted terms were manually evaluated for their correctness, i.e., their membership in the corresponding category. There are three annotators (who were not authors), two of which are native speakers. We use the majority vote as the ground truth.

Baselines: we compare our system against three baselines: (a) *OurApproach – Expansion*: this baseline removes lines 4-5 in Algorithm 1 and uses the known terms ($\mathcal{E}$) rather than the expanded terms (E) for pattern ranking (Eq. 4); (b) *OurApproach – Polarity*: this model removes all polarity information: only the positive category is used in Eqs. 1, 2, 3, 4, and 6, and no cautionessness constraints in Section 3.1. and Section 3.3.; and (c) *McIntosh++*: our implementation of the mutually-exclusive approach of McIntosh and Curran (2008): no term expansion, no polarity, and categories are required to be mutually exclusive. Our extension to this system was to add log(frequency) to their ranking formula based on MI^2 (similar to Eq. 4 and 6); we found this performs better on our small data.

Hyper parameters: there are three hyper parameters in our approach: λ_1, λ_2 , and λ_3 (see Section 3.). Intuitively, we aim for a relatively high value for λ_1 because we prefer fewer, high-quality expanded terms (i.e., the expansion should be cautious). We aim for relatively low values for λ_2 and λ_3 because these are coupled with additional constraints in Section 3. that reduce the risk of introducing noise. We tuned all these parameters on the training dataset, and found that performance is best when $\lambda_1 = 0.5$, $\lambda_2 = 0.2$, $\lambda_3 = 0.1$.

Results: Figure 2 plots the cumulative term precision and throughput (i.e., number of terms learned in a given category) for our approach and the three baselines, for four emotion dimension categories. The figure shows that all our contributions are important. Term expansion yields a considerable improvement in both precision and throughput, especially for the negative emotion dimensions, which are more affected by sparsity. Polarity information consistently improves precision for all categories. Our approach has considerable higher precision than McIntosh++, at a small loss in throughput for two categories.

An ablation test (shown in Figure 3) of the resources used indicates that WordNet has the highest contribution to precision (e.g., yielding an increase of 15% (absolute) for *blameworthy*, and 20% for *undesirable*), and `word2vec` has the highest contribution to throughput (e.g., yielding an increase from 80 to 145 terms for *praiseworthy*). Edit distance has a small contribution to precision for *undesirable* and *desirable*, confirming that misspellings occur in informal texts, but it impacts negatively the *blameworthy* category, suggesting that it can also accumulate er-

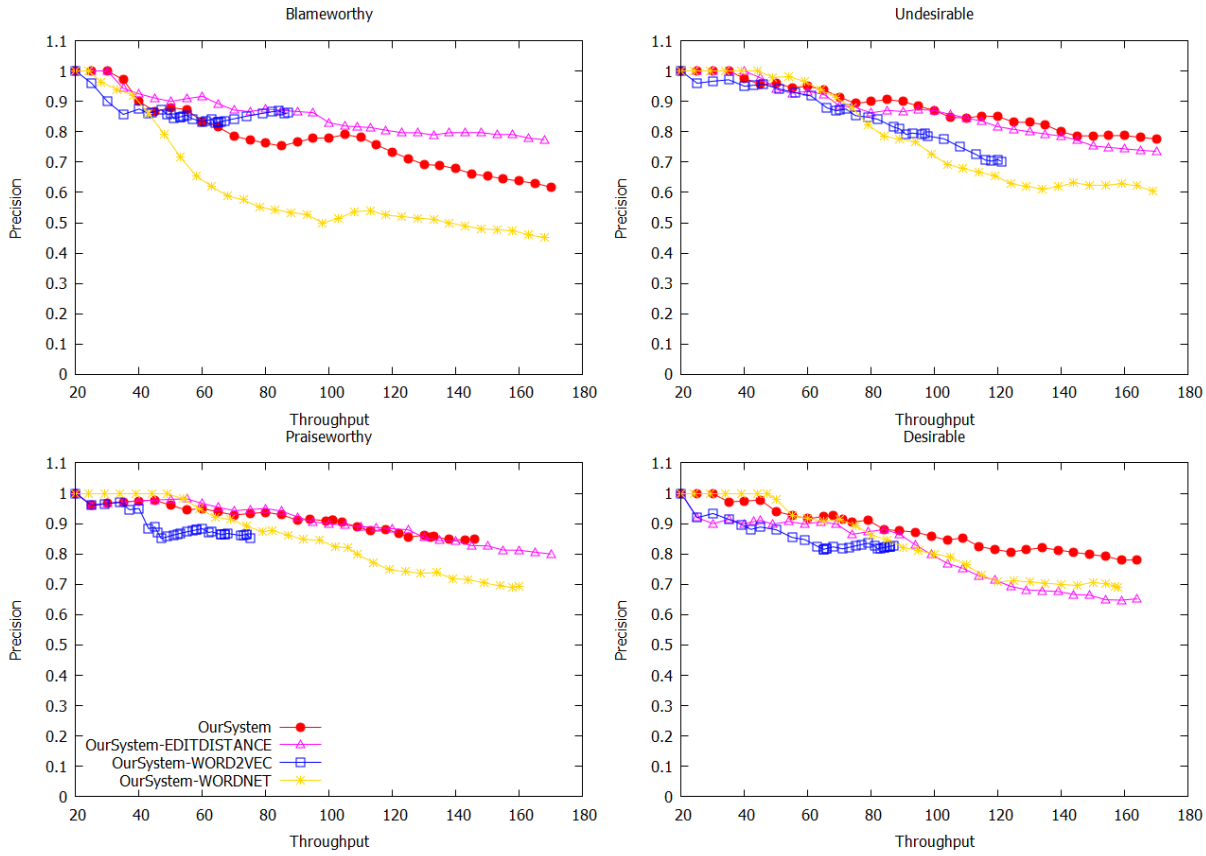

Figure 3: Ablation test for the three resources used in our approach: edit distance, `word2vec`, and WordNet. All systems were run for up to 30 epochs.

rors, e.g., by generating multiple spellings of terms that are incorrect.

5. Conclusion

We introduced a novel bootstrapping approach for the extraction of lexicons for polarized categories, in our case emotion dimensions of the OCC cognitive model of emotions (Ortony et al., 1990). We focused on small datasets containing informal texts, and made several contributions. First, we mitigated the sparsity of the data with a term expansion component that takes advantage of multiple resources: WordNet, word embeddings, and edit distance, and showed that these resources used have complementary contributions. Second, we addressed the semantic drift limitation of bootstrapping with multiple solutions that take advantage of the polarity of the classes to be learned. Our approach yields considerable higher precision than a traditional counter-training system (in some cases more than double), with only a small loss of throughput for some categories.

6. Bibliographical References

Blum, A. and Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM.

Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Jr., E. R. H., and Mitchell, T. M. (2010). Toward an architecture for never-ending language learning. In *AAAI*.

Collins, M. and Singer, Y. (1999). Unsupervised models for named entity classification. In *EMNLP*.

Gupta, S. and Manning, C. D. (2014). Improved pattern learning for bootstrapped entity extraction. In *CoNLL*.

Gupta, S. and Manning, C. D. (2015). Distributed representations of words to guide bootstrapped entity classifiers. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*.

Kozareva, Z. and Hovy, E. (2010). A semi-supervised method to learn and construct taxonomies using the web. In *EMNLP*.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). The stanford corenlp natural language processing toolkit. In *ACL*.

McIntosh, T. and Curran, J. R. (2008). Weighted mutual exclusion bootstrapping for domain independent lexicon and template acquisition. In *Proceedings of the Australasian Language Technology Association Workshop*.

McIntosh, T. (2010). Unsupervised discovery of negative categories in lexicon bootstrapping. In *EMNLP*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In *ICLR Workshop*.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. J. (1990). Introduction to wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–244.

Ortony, A., Clore, G. L., and Collins, A. (1990). *The Cognitive Structure of Emotions*. American Psychological Association.

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

Riloff, E. (1996). Automatically generating extraction patterns from untagged text. In *AAAI*.

Yangarber, R. (2003). Counter-training in discovery of semantic patterns. In *ACL*.

Yih, W., Zweig, G., and Platt, J. C. (2012). Polarity inducing latent semantic analysis. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.