

Mining Slang and Urban Opinion Words and Phrases from cQA Services: An Optimization Approach

Hadi Amiri and Tat-Seng Chua

NUS Graduate School for Integrative Sciences and Engineering
Department of Computer Science, School of Computing
National University of Singapore

{hadi,chuats}@comp.nus.edu.sg

ABSTRACT

Current opinion lexicons contain most of the common opinion words, but they miss slang and so-called urban opinion words and phrases (e.g. *delish*, *cozy*, *yummy*, *nerdy*, and *yuck*). These subjectivity clues are frequently used in community questions and are useful for opinion question analysis. This paper introduces a principled approach to constructing an opinion lexicon for community-based question answering (cQA) services. We formulate the opinion lexicon induction as a semi-supervised learning task in the graph context. Our method makes use of existing opinion words to extract new opinion entities (slang and urban words/phrases) from community questions. It then models the opinion entities in a graph context to learn the polarity of the new opinion entities based on the graph connectivity information. In contrast to previous approaches, our method not only learns such polarities from the labeled data but also from the unlabeled data and is more feasible in the web context where the dictionary-based relations (such as *synonym*, *antonym*, or *hyponym*) between most words are not available for constructing a high quality graph. The experiments show that our approach is effective both in terms of the quality of the discovered new opinion entities as well as its ability in inferring their polarity. Furthermore, since the value of opinion lexicons lies in their usefulness in applications, we show the utility of the constructed lexicon in the sentiment classification task.

Categories and Subject Descriptors

I.2.7 [Natural Language Processing]: Text Analysis; H.3.3 [Information Search and Retrieval]: Text Mining

General Terms

Algorithms, Experimentation.

Keywords

Slang, Urban Word, Opinion Lexicon, Sentiment Orientation, Opinion Mining, Sentiment Analysis.

1. INTRODUCTION

Community-based Question Answering (cQA) services are web portals that allow users to post questions, respond to the previously asked questions, and rate/vote answers, etc. Well-

known examples of such services are Yahoo! Answers¹, AnswerBag², and Baidu Zhidao³. The cQA questions usually contain a high amount of spelling errors and usage of slang. These issues make the analysis of such questions very difficult. Hence, the cQA data are triggering many innovative research scenarios in areas as diverse as user satisfaction [1], question recommendation [32], answer ranking and selection [23], and opinion summarization [27].

In this research we focus on the more fundamental issue of constructing an opinion lexicon from cQA questions. Opinion lexicons contain opinion words with their polarity labels, either positive or negative, and some scores that represent the strength of their polarity. Opinion lexicons are essential resources for almost all the tasks of sentiment analysis such as opinion mining [8], opinion retrieval [9], and opinion question answering [14]. In particular, in the cQA context, opinion lexicons can be used for discriminating opinion from factual questions [13], and answer summarization [27]. Given a question, a good cQA opinion lexicon can help determining the question as opinion or factual and deciding whether we should employ opinion-specific summarization methods in generating the answer or just rely on the traditional summarization techniques [2, 3].

There exist three approaches to opinion lexicon construction: manual, dictionary-based, and corpus-based approaches [19, 15]. The manual approach needs human annotators to tag the individual words as positive, negative or neutral. Dictionary-based approaches use a set of seeds (opinion words with already-known polarity) and search their synonyms and antonyms in dictionaries like WordNet⁴ to find new opinion words [6, 10, 8, 24, 22, 21, 17]. An important shortcoming of the dictionary-based approaches is the limited vocabulary or coverage problem. In fact, the dictionary-based methods cannot find informal or so-called urban opinion words due to the fact that such words do not appear in the dictionaries. To address this problem, the corpus-based approaches are evolved that use synthetic or co-occurrence patterns in the text for constructing opinion lexicons [7, 11, 28, 29, 4]. The corpus-based approaches can help to find domain and context specific opinion words or phrases and their polarity orientations using a domain corpus. We review some of these works in the next section.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WSDM'12, February 8–12, 2011, Seattle, Washington, USA.
Copyright 2012 ACM 978-1-4503-0747-5/12/02...\$10.00.

¹ <http://answers.yahoo.com/>

² <http://www.answerbag.com/>

³ <http://zhidao.baidu.com/>

⁴ WordNet: <http://wordnet.princeton.edu/>

In this research we divide the cQA opinion lexicon construction task into two subtasks:

1. New Opinion Entity Detection: To the best of our knowledge, there is no principled approach to detect new *opinion entities* (words or phrases). Previous research either: (1) designed hand-crafted rules [7, 21, 11, 20], or (2) used dictionaries and WordNet relations [8, 26, 6, 22, 12] for this purpose. Each of the above two approaches has its own advantages and disadvantages. For instance both rule-based and dictionary-based approaches are precise, but rules are hard to design and dictionaries have limited vocabulary (coverage) problem. In this research, we propose a principled approach to detect new opinion entities in the cQA context. Our approach effectively combines the above-mentioned methods in a unified framework and is able to detect non-standard entities such as urban opinion words/phrases, slang and misspellings etc.

2. Polarity Inference: The association between seeds and new opinion entities provides a rich source of relationships. We model such relationships in a graph context to assign polarity to new opinion entities. Most of the previous methods only utilized labeled data (seeds) to predict such polarities [e.g. 28, 11], or used synonym, antonym, or hyponym relations available in dictionaries to construct a high quality graph [10, 24, 22, 17]. In contrast, we make use of both labeled and unlabeled data to predict the polarities, and construct the graph from the *polarity association* between the words. Our polarity inference method is more feasible in the Web context where the data contains many non-standard entities and the above dictionary-based relations are not available. We formulate the polarity inference task as a semi-supervised learning task in the graph context where the seeds and opinion entities are modeled as the graph nodes and the aim is to optimize the polarity of new opinion entities based on the graph connectivity information. Similar to [28, 4, 29] we use co-occurrence as the polarity association measure. This is because previous research shows that opinion words with opposite orientation tend not to co-occur in the same context, while opinion words with similar orientation tend to co-occur [4].

To summarize, the contributions of this paper are as follows:

- We propose a principled approach to mine new opinion words in the cQA context. Our approach can be used as a feature selection method for mining sentiment terminology.
- We present a novel adaptation of graph-based semi-supervised learning methods to learn the polarity of new opinion entities in the graph context.
- Our opinion lexicon contains non-standard entities such as urban opinion words/phrases, slang and misspellings etc that cannot be found in the current popular opinion lexicons.

The experiments show that our polarity inference method significantly improves the performance of the two baselines by 21.16% and 2.78% in F1 score. We also investigate the effect of unlabeled data in polarity inference and show that the above improvement comes from the optimization framework and learning from unlabeled data. In addition, the sentiment classification experiments show that our method is effective in detecting new opinion entities and constructing a high quality cQA opinion lexicon. Our best performing method significantly improves the performance of the baseline (seeds) by 5.89% in F1 score.

The rest of this paper is organized as follows. Section 2 surveys the related work. Section 3 gives an overview of our approach. Section 4 elaborates our method for mining new opinion entities and explains some linguistic considerations for this purpose. Section 5 describes our optimization framework for polarity inference. Section 6 reports the experimental settings and results on both polarity inference and sentiment classification tasks, and, finally, Section 7 concludes the paper and discusses future directions and plans.

2. RELATED WORK

As aforementioned, there are three approaches to opinion lexicon construction: manual, dictionary-based, and corpus-based approaches [19, 15].

As a dictionary-based approach for opinion lexicon construction, Hu and Liu [8] proposed to consider the synonyms and antonyms of seeds in WordNet as new opinion words. The synonyms and antonyms were accumulated in an opinion lexicon and the process was repeated until no new word could be added to the lexicon. Qiu et al. [21] used a similar method as [8], but their algorithm also exploits the relations between sentiment words and product features that the sentiment words modify. Kamps et al. [10] used WordNet to construct a graph of synonyms. They used the shortest paths from a given word to the seeds "*good*" and "*bad*" to determine the polarity of the word. They reported that in WordNet synonymy graph, the words "*good*" and "*bad*" themselves are quite close to each other and concluded that the shortest path in such a graph could be noisy. Rao and Ravichandran [22] used synonym and hyponym of seeds to construct the graph and utilized the label propagation technique proposed in [5] to propagate the polarity labels in the graph.

In contrast to dictionary-based approaches, corpus-based approaches use synthetic or co-occurrence patterns in the text. As a corpus-based approach, Hatzivassiloglou and McKeown [7] and Kanayama and Nasukawa [11] used seed opinion adjectives with conjunctions like "*and*", "*or*", "*but*", "*either-or*", and "*neither-nor*" to find more seeds. For example, "*and*" was used as an evidence of the same polarity ("*cheap and comfortable*"). So, if we know the polarity of one of the words in a conjunctive expression, we can deduce the polarity of the other word using the above clues. Kanayama and Nasukawa [11] further expanded the above heuristics to consider the proximity between the sentences in reviews based on the assumption that the same opinion orientation is usually expressed in a few consecutive sentences. The conjunction-based rules have been used in many research works as an aid to help developing opinion lexicons [26, 20].

Turney [28] and Turney and Littman [29] proposed to determine the polarity of a word/phrase by comparing whether it has a greater tendency to co-occur with positive opinion words (e.g. "*excellent*") or negative opinion words (e.g. "*poor*"). In particular, given a word w , they determined the polarity of w as the point-wise mutual information (PMI) between w and a fixed set of positive words minus the PMI between w and a fixed set of negative words. This approach is simple and feasible, but it does not guarantee that two highly co-related opinion words will be assigned the same polarity. In this paper, we utilize this approach as a feature selection method for mining opinion terminology and improve it by utilizing unlabeled data in an optimization framework.

Constructing opinion lexicons from the web content has been studied previously in [18] and [30]. Kaji and Kitsuregawa [18] focused on Japanese text and proposed a method to construct an

opinion lexicon that only contains adjective phrases. They utilized seeds and HTML structural clues (such as *bulleted lists* and *tables*) to first extract opinion sentences from web pages. They then used a syntactic parser to extract opinion adjective phrases from the sentences. The polarity of a phrase was then determined using Turney's method [28]. In contrast, our research focus is on extracting any type of opinion entities from the cQA context where the only available information is the question threads.

Velikovich et al. [30] proposed a Graph Propagation (GP) technique to perform polarity inference in the graph context. They considered word n-grams as the nodes of a graph and weighted edges based on the cosine similarity between the context features of their nodes (extracted from Web n-grams for each node). They computed both positive and negative scores for each unlabeled node based on the maximum weighted paths between the node and all seeds. The polarity of the unlabeled node was then computed based on Turney's method, i.e. the difference between the two positive and negative scores. We show that our polarity inference approach better utilizes the unlabeled nodes by constructing a *smooth* polarity graph and outperforms the GP method in the polarity inference task.

3. OVERVIEW OF OUR APPROACH

This section presents an overview of our approach for constructing the cQA opinion lexicon. We construct the lexicon in two steps: (1) mining candidate opinion entities, and (2) inferring the polarity of the entities.

Step 1: We first extract a set of candidate opinion entities using seeds (the words with already-known polarity). Having two classes of positive and negative seeds, we extract entities (words or phrases) that frequently co-occur with one class (e.g. positive seeds) and rarely with its opposite class (e.g. negative seeds). We expect these entities to be rich in sentiment. We refer to such entities as *Significant Entities* (SEs) and consider them as candidate opinion entities. For instance "*coool place*" and "*recommend*" are SE because they frequently co-occur with seeds like *fun*, *favorite* etc and rarely with *bad*, *terrible* etc. However, the entity "*to go*" (semi) equally co-occurs with both positive and negative seeds and cannot be a SE.

Step 2: In the next step, we construct a *polarity graph* from the seeds and the extracted SEs as depicted in Figure 1. In this Figure, the '+' and '-' nodes are labeled nodes (positive and negative seeds respectively), and the '?' nodes are SEs or unlabeled nodes. Each SE node is attached to a corresponding d-node (the black nodes) that contains an initial polarity prediction for the SE. The initial predictions are optional. We explain d-node prediction in Section 5.3. The solid edges in the graph reflect the *polarity association* between the nodes. The weight of such edges is computed as a function of the co-occurrence between their corresponding nodes. In our polarity graph, we restrict such edges to happen only between SEs and seeds, and any two seeds with the same polarity. This prevents the opposite seeds from directly propagating their labels through each other. Once the graph is constructed, the polarity inference problem can be best modeled as a semi-supervised learning task in the graph context where the labeled nodes are the seeds and the unlabeled nodes are SEs and the aim is to optimize the polarity of SEs based on the graph connectivity information. We treat the SEs with sufficiently high confidence as new opinion entities and add them to the cQA opinion lexicon.

The above two Steps construct the polarity graph without using any dictionary or dictionary-based relations between the nodes. As such, our method is more feasible in the Web context where

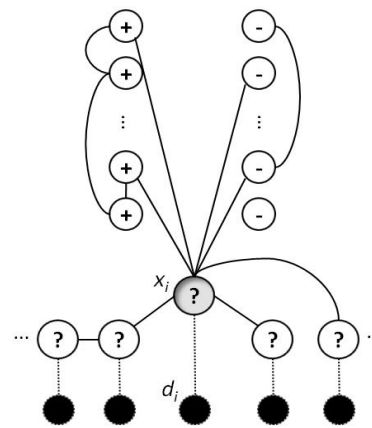


Figure 1. The connectivity information available in the polarity graph. '+' and '-' indicate positive and negative seeds respectively, '?' indicates SEs, and the black nodes are the initial polarity predictions for SEs.

such relations are generally not available among many non-standard entities.

4. MINING SIGNIFICANT ENTITIES

In this section, we aim to extract significant entities (SEs) and treat them as candidate opinion entities. We first explain some linguistic considerations and then describe our method for mining SEs.

4.1 Linguistic Considerations

We first utilize three sources of easy-to-collect information as seeds. These sources provide high quality seeds that have high confidence (precision) but low coverage (recall) in sentiment:

- **General Purpose Opinion Lexicons:** We consider those opinion words as seeds that are either labeled as strong in the General Inquirer [25] or subjectivity lexicon [33], or have positive or negative score of one in SentiWordNet [24]. For SentiWordNet, we only consider the first sense of the words.
- **Linguistic Rules:** As we mentioned before, previous research designed linguistic rules to detect more opinion words. For example, the affixes "*dis*" and "*mis*" were used as evidences for opposite polarities (*honest* ↔ *dishonest*, *fortune* ↔ *misfortune*). We use the above seeds and linguistic rules of [17] to find more high quality seeds.
- **WordNet Similarity:** We extract the synonym and antonym of each seed from WordNet and consider them as seeds too. The synonyms will be assigned the same polarity as their corresponding seeds, while the antonyms will receive the opposite polarity. We do not repeat this process because we want to ensure the high confidence (precision) of seeds.

The above sources provide an initial set of seeds. We only consider the seeds that occur more than once in our development corpus. These seeds will then be used to mine the significant entities.

In addition we found that *Negations* and *Disjunctive Clauses* are important factors to appropriately relate entities to seeds. For

example, if a seed is negated, its context words tend to co-occur with the seed's antonym but not the seed itself. Sub-sentences of a disjunctive clause⁵ also have opposite polarities. We explain below the way we handle negations and disjunctive clauses in detail:

Negations: Negation words/phrases such as *not*, *none*, *cannot*, *barely*, *lack of*, and *never* etc reverse the sentiment of the seed words. Parser toolkits are useful resources to detect negations and their dependencies in the text. We consider the *clause* that contains a negation word as the scope of the negation. However, because of the weak grammar and the presence of high amount of short-form texts in cQA data, we designed some manual rules to better tackle the negation. We also consider cases that the negation word is not negating the seeds such as "*not only ... but also ...*", "*last but not least ...*" etc. In total we compiled 36 negation words and rules.

Disjunctive Clauses: We consider disjunctions like *but*, *though*, *although*, *despite*, *in spite of*, *except for*, *except that* etc to relate the entities and seeds. Consider an opinion sentence with two clauses connected by the disjunction "*but*", such as "*CLAUSE1, but CLAUSE2*" where *CLAUSE1* contains the seed word *s*. These two clauses should have opposite sentiments because of the disjunction "*but*". Therefore, we can say that the entities of *CLAUSE2* co-occur with the antonym of *s* instead of *s*. For example given the sentence: "*I think it's stylish to hang artworks on walls, but nowadays it's kind of tacky to hang up posters!*" with the word "*stylish*" as its seed, the entities in the clause after "*but*" such as "*tacky*", "*it's kind of tacky*", etc should be related to the antonym of "*stylish*". We also designed a few manual rules to better detect and handle disjunctive clauses. We utilize Stanford toolkit to extract clauses and split questions and answers into sentences.

4.2 Significant Entity Extraction

As aforementioned, our assumption is that the entities that frequently co-occur with positive seeds and rarely (or never) with negative seeds are highly likely to be positive. Similarly, the entities that frequently co-occur with negative seeds and rarely (or never) with positive seeds are highly likely to be negative [28, 29, 30, 18].

We use Point-wise Mutual Information (PMI) as the measure of co-occurrence. PMI between two words *v* and *w* is defined as follows:

$$PMI(v, w) = \log\left(\frac{P(v \& w)}{P(v)P(w)}\right) = \log\left(\frac{Count(v \& w)M}{Count(v)Count(w)}\right) \quad (1)$$

where $P(v \& w)$ is the probability that the two words co-occur in the same context (e.g., a sentence or several consecutive sentences), and $P(v)$ and $P(w)$ are the probability of *v* and *w* occurring in the entire corpus respectively, and M is a constant (e.g. the number of terms in the corpus). PMI is a good measure to associate the words that frequently co-occur in the same context [29, 35].

We define an entity as any word *N-grams* ($N = 1, 2, \text{ or } 3$) extracted from the cQA questions or their answers. Our aim is to use seeds to find SEs. For this purpose, for each seed s_i , we

⁵ A disjunctive clause contains two sub-sentences connected by a disjunction (e.g. *but*).

extract all the entities that occur in the context of s_i , compute their PMI with respect to s_i , and accumulate them in set N_{s_i} as the set of neighboring entities of s_i .

We consider the sentence that contains s_i and its previous sentence as the context of s_i . It is necessary to consider a set of consecutive sentences as the context of the seed words for the following 2 reasons: (1) many new opinion entities do not co-occur with any seed word at the sentence level, and (2) the same opinion orientation is usually expressed in a few consecutive sentences [11]. So we can expect the same orientation among the entities of consecutive sentences. However, we limit the above requirement as follows: (1) if the previous sentence contains a seed with opposite polarity with s_i , we do not consider that sentence in the context of s_i ; and (2) if the current sentence contains two seeds with opposite polarities, we only consider the previous sentence as the context of the seed that appears first.

We create the entity pool N from the sets of neighboring entities as follows:

$$N = \bigcup_i N_{s_i} \quad (2)$$

We then compute an *initial polarity score* for each entity $e_k \in N$. This score is computed as a function of entity's co-occurrence with positive and negative seeds as follows:

$$InitPScore_{e_k} = \sum_{s_i \in Pos} PMI(s_i, e_k) - \sum_{s_j \in Neg} PMI(s_j, e_k) \quad (3)$$

where *Pos* is the set of positive seeds and *Neg* is the set of negative seeds. In Equation (3), we only consider positive PMI values because it reflects positive co-relation between entities and seeds. The above Equation measures the tendency of the entities towards positive or negative classes of seeds. In the above Equation, $|InitPScore_{e_k}|$ will be high for entities that are highly associated with only one of the positive or negative classes. We first normalize the *InitPScores* and then sort the entities in descending order of the *absolute* values of their *InitPScores*. We then pick the *Top K* entities from this set and consider them as significant entities. These SEs are expected to be rich in sentiment.

5. POLARITY INFERENCE

The SEs and seeds provide a rich source of relationships. Such information can be readily encoded as a graph where the presence of an edge represents a relationship between the nodes (here polarity association) and the weight of the edge represents the strength of the relationship. This information can help to learn the polarity of candidate opinion entities when only a few seeds are present.

Formally, the polarity inference problem can be described as follows:

Assume that there exist n instances $\{x_1, \dots, x_n\}$ in the dataset \mathcal{X} . Let the first l instances $\mathcal{X}^l = \{x_1, \dots, x_l\}$ be the labeled data (seeds) and the remaining instance $\mathcal{X}^u = \{x_{l+1}, \dots, x_n\}$ be the unlabeled data (the SEs). Let y_i indicates the label (polarity score) of x_i ; with $y_i = +1$ for positive seeds and $y_i = -1$ for the negative seeds. The aim is to find a real-valued function $f: x \rightarrow \mathbb{R}$ that gives a continuous polarity score $f(x)$ to entity x . The value of function f on the labeled data x_i is the same as its initial label y_i , i.e. $f(x_i) = y_i$ for $i = 1, \dots, l$. The problem is then to determine the polarity of the unlabeled nodes, i.e. $f(x_j), j = l+1, \dots, n$.

This problem can best be modeled as a semi-supervised learning task in the graph context where the connectivity information of the graph can be utilized to estimate the polarity scores for the unlabeled data. We first construct the polarity graph from the SEs and seeds and then define the optimization criteria.

5.1 Polarity Graph Construction

Let $G=(V, E)$ be an undirected edge-weighted graph defined on the dataset \mathcal{X} with nodes V corresponding to the n entities of \mathcal{X} , and edges E that are weighted by an $n \times n$ symmetric weight matrix \mathbf{W} . The weight of the edge $(v_i, v_j) \in E$, w_{ij} , indicates the polarity association between the nodes v_i and v_j and is obtained from $\text{PMI}(v_i, v_j)$ as defined in Equation (1). Formally, we construct G as follows:

- Any $x_j \in \mathcal{X}^u$ is connected to all the \mathcal{X}^u and \mathcal{X}^l nodes that have positive PMI with x_j , and
- Any $x_i \in \mathcal{X}^l$ is connected to all the \mathcal{X}^l nodes that have the same polarity and positive PMI with x_i .

If there is no edge between two nodes its corresponding weight is deemed to be 0. Note that the PMI function is a *symmetric function*, i.e. $\text{PMI}(a, b) = \text{PMI}(b, a)$. The above configuration results in a large graph in which each unlabeled nodes (SE) is potentially connected to several labeled nodes and other unlabeled nodes through different edges/paths (see Figure 1).

Furthermore, we assume that, we have an initial polarity prediction (also called *dongle* node [34]) for each unlabeled node, i.e. $f(x_i) = \hat{y}_i$ for $i=l+1 \dots n$. Each d-node is connected to its corresponding unlabeled node with the edge weight of 1 and acts as prior knowledge for the semi-supervised learning framework. \hat{y}_i is set to zero when there is no initial prediction. We explain how to estimate the value of d-nodes in section 5.3.

5.2 Optimization Framework

The basic idea of the semi-supervised learning algorithms in the graph context is that the function $f(x)$ should be *smooth* with respect to the graph [5, 34, 31]. $f(x)$ is not smooth with respect to the graph if there is a heavy edge with weight w_{ij} between two nodes x_i and x_j , and the difference between $f(x_i)$ and $f(x_j)$ is large, i.e. $w_{ij} (f(x_i) - f(x_j))^2$ is large. Therefore, the aim of the optimization framework is to minimize the above value over all the edges in the polarity graph.

Assuming that the d-nodes in Figure 1 are connected to their corresponding unlabeled nodes with the weight of 1, our aim is to minimize the following *energy* function:

$$E(f) = \gamma \sum_{x_i \in \mathcal{X}^u} (f(x_i) - \hat{y}_i)^2 + (1 - \gamma) \sum_{x_i \in \mathcal{X}^u} \left(\sum_{x_j \in \text{Adj}^l(x_i)} \alpha w_{ij} (f(x_i) - f(x_j))^2 + \sum_{x_j \in \text{Adj}^u(x_i)} (1 - \alpha) w_{ij} (f(x_i) - f(x_j))^2 \right) \quad (4)$$

where $\text{Adj}^l(x_i)$ and $\text{Adj}^u(x_i)$ are the sets of x_i 's adjacent labeled and unlabeled nodes respectively, the parameter $\gamma \in [0,1]$ represents the influence of each source of learning (dongle node

vs. adjacent nodes) on the polarity of x_i , and the coefficient $\alpha \in [0,1]$ controls the effect of labeled and unlabeled nodes on the polarity of the unlabeled node x_i . Equation (4) represents the requirements that for each unlabeled node $x_i \in \mathcal{X}^u$, we want $f(x_i)$ to be consistent with its d-node, and its neighbors. The greater values of α reduce the effect of the adjacent unlabeled nodes on the polarity of x_i , while smaller values of α increase such effects. Since the paths from unlabeled nodes could potentially be noisy, we set $\alpha \geq 0.5$ to produce better performance.

The optimization problem can be defined as follows:

$$\hat{\mathbf{f}} = \underset{f}{\text{argmin}} E(f) \quad (5)$$

To find a closed-form solution to the above Equation, we define an $n \times n$ matrix \mathbf{T} as follows:

$$T_{ij} = \begin{cases} 0, & i \in L, j \in L \\ \alpha(1 - \gamma)w_{ij}, & i \in L, j \in U \\ \alpha(1 - \gamma)w_{ij}, & i \in U, j \in L \\ 2(1 - \alpha)(1 - \gamma)w_{ij}, & i \in U, j \in U \end{cases} \quad (6)$$

where $L=1 \dots l$ and $U=l+1 \dots n$ are the labeled and unlabeled node indices respectively. Let \mathbf{D} be a diagonal matrix derived from \mathbf{T} as follows:

$$D_{ii} = \sum_{j=1}^n T_{ij} \quad (7)$$

Let $\mathbf{\Omega} = \mathbf{D} - \mathbf{T}$ be the $n \times n$ *graph Laplacian* matrix [16], $\mathbf{f} = [f(x_1), \dots, f(x_n)]^T$, and $\mathbf{y} = [y_1, \dots, y_l, \hat{y}_{l+1}, \dots, \hat{y}_n]^T$ where $f(x_i) = y_i$ for the labeled nodes ($i=1 \dots l$), and \hat{y}_j is the value of the dongle nodes for the unlabeled nodes ($j=l+1 \dots n$). We can rewrite Equation (4) as follows where \mathbf{I} is the $n \times n$ identity matrix (see Appendix A for the derivations):

$$E(f) = \gamma(\mathbf{f} - \mathbf{y})^T \mathbf{I} (\mathbf{f} - \mathbf{y}) + \mathbf{f}^T \mathbf{\Omega} \mathbf{f} \quad (8)$$

The minimum energy function $\hat{\mathbf{f}}$ of the above quadratic function can be obtained as follows:

$$\frac{\partial E(f)}{\partial \mathbf{f}} = 0 \Rightarrow \hat{\mathbf{f}} = \gamma(\gamma \mathbf{I} + \mathbf{\Omega})^{-1} \mathbf{y} \quad (9)$$

Because $\mathbf{f}^T \mathbf{\Omega} \mathbf{f} > 0$, $\mathbf{\Omega}$ is a symmetric and positive semi-definite matrix and consequently the above solution is the unique answer to our optimization problem. We normalize this vector into $[-1, 1]$ range.

5.3 Polarity Prediction for Dongle Nodes

Given an unlabeled node, $x_i \in \mathcal{X}^u$, we utilize two methods to predict the polarity value of its corresponding d-node, d_i . The first intuitive method is based on Equation (3) that computes an initial polarity prediction for the unlabeled nodes. We refer to this prediction as CO in the experiments.

As the second prediction method, we make use of the idea proposed in [30]. In particular, we compute a positive and a negative score for each unlabeled node $x_i \in \mathcal{X}^u$. The positive

score is computed as the sum over the maximum weighted path from every positive labeled node to x_i . Similarly, the negative score is computed as the sum over the maximum weighted path from every negative labeled node to x_i . The value of the corresponding d-node is then computed as the difference between the two positive and negative scores. Mathematically, for each node $x_i \in \mathcal{X}^u$, we compute the value of its d-node as follows:

$$d_i = \frac{1}{Z} \left(\sum_{x_j \in Pos} S_{ij} - \varphi \sum_{x_k \in Neg} S_{ik} \right) \quad (10)$$

where *Pos* and *Neg* are the positive and negative labeled nodes in \mathcal{X}^l respectively, Z is a normalization term, S_{ij} is the value of the maximum weighted path from x_i to x_j , and φ is a constant value that accounts for the difference in the overall mass of positive and negative flow in the graph, and is computed as follows:

$$\varphi = \frac{\sum_{x_i \in \mathcal{X}^u} \sum_{x_j \in Pos} S_{ij}}{\sum_{x_i \in \mathcal{X}^u} \sum_{x_j \in Neg} S_{ij}} \quad (11)$$

Equation (10) assigns high positive (negative) scores to unlabeled nodes that are connected to multiple positive (negative) labeled nodes through short yet highly weighted paths [30]. If x_i has higher positive score than negative score, then its initial guess will be positive, i.e. $d_i > 0$; and $d_i < 0$ otherwise. We refer to this prediction as GP in the experiments.

Our optimization framework improves upon these two baselines by: (1) imposing the smoothness restriction over the polarity graph (see Section 4.2.), and (2) preventing label propagation through seeds with opposite polarities.

6. EXPERIMENTS

In this Section we evaluate our approach from two perspectives:

Polarity Inference: We first evaluate the ability of the optimization framework in inferring the polarity of opinion words (polarity inference). We utilize the seed opinion words for this purpose. We assume that part of the seed dataset is unlabeled and evaluate the performance of our optimization framework in predicting the correct label of such seeds.

Sentiment Classification: We then evaluate the quality of the extracted new opinion entities. For this purpose, similar to [30, 28], we consider a word-matching-based review classification task as the measure of evaluation. We expect opinion entities with higher quality result in higher performance of review classification

6.1 Data

We used the newly released Yahoo! Webscope dataset⁶ as the development dataset for mining opinion entities. This dataset is collected from Yahoo! Answers cQA archives (as of 10/25/2007). We considered each question and each answer as an individual document and performed the experiments on the *Food* domain. The *Food* domain of this collection contains 244K documents and 0.5M sentences. We use these documents to detect SEs and extract co-occurrence information.

From the seed words that we compiled in Section 4.1, we only kept the seeds that occur more than once in the cQA corpus. In

⁶ <http://webscope.sandbox.yahoo.com/>

this way, we obtained more than 2,500 seeds (almost balanced on positive and negative categories).

For the purpose of the second evaluation, we used a restaurant review dataset crawled from *newyork.citysearch.com*⁷. In this dataset, each review has a rating star scaling from 1 (highly negative) to 5 (highly positive). We used a balanced set of positive and negative reviews for the evaluation purpose (7000 on positive and 7000 on negative reviews).

All the experiments in the subsequent Sections were performed through 10-fold cross validation and the two-tailed paired t-test $p < 0.01$ was used for significance testing. Throughout this Section, we use the asterisk mark (*) to indicate significant improvement over the best performing baseline.

6.2 Polarity Inference Performance

We use the seed dataset as the ground-truth to evaluate the performance of our optimization framework in polarity inference. For this purpose, we consider part of the seed dataset as the test data (unlabeled nodes) and the rest of the seeds as training data (labeled nodes), and evaluate the performance of the optimization framework in predicting the polarity of the unlabeled nodes. We use the following measures for the evaluation:

$$Precision = \frac{N_{correct}}{N_{tagged}}$$

$$Recall = \frac{N_{correct}}{N_{unlabeled}}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

where $N_{correct}$ is the number of unlabeled nodes that are assigned the correct polarities (either positive or negative), N_{tagged} is the number of unlabeled nodes that are assigned non-zero scores, and $N_{unlabeled}$ is the total number of unlabeled nodes.

We use 80% of the seed dataset for training and the rest for testing. In addition, we use 10% of the training set to tune the parameter α . For this purpose, we employ a greed search in the [0.1, 1] range with the greed step of 0.1. We analyze the effect of this parameter on the performance of polarity inference in the next Section. In addition, We treat the dnode nodes as other nodes in the graph and empirically set the value of γ to 0.5⁸.

Table 1 presents the results. CO indicates the results when we use the co-occurrence information (Equation 3) to predict the polarity labels of test data. As Table 1 shows, CO produces a low F1 performance of 47.89%. This poor performance is due to the fact that CO ignores the co-occurrence among the unlabeled data. However, we should mention that the performance of CO highly depends on the amount of raw text provided for computing the co-occurrence information.

As Table 1 shows, GP produces a higher F1 performance than CO

⁷ The restaurant review dataset has been obtained from <http://www.cs.cmu.edu/~mehrbod/RR/>.

⁸ We also experimented with some other values of γ and observed that giving more weight to adjacent nodes improves the performance when there is no prediction available for the d-nodes.

Table 1. Performance of Polarity Assignment for Different Methods

Methods	Precision	Recall	F1
CO	47.89	47.89	47.89
GP	66.27	66.27	66.27
OPT, $\alpha: .5$	69.45	65.06	67.19*
OPT-CO, $\alpha: .7$	65.59	61.45	63.45
OPT-GP, $\alpha: .7$	71.38	66.87	69.05*

(66.27% vs. 47.89%). This difference is significant and stems from GP’s utilization of both edges (direct co-occurrence) and paths (indirect co-occurrence) of the polarity graph. We consider GP and CO as the baselines.

The results of the optimization framework are shown in the last three rows of Table 1. OPT indicates the result when there is no initial predictions for the unlabeled nodes, i.e. $\hat{y}_i = 0$ for $i=l+1 \dots n$. As it is shown, it outperforms the CO and GP methods significantly and produces a F1 performance of 67.19%. OPT, in contrast to CO or GP, optimizes the polarity of unlabeled nodes by imposing the smoothness restriction on the polarity graph. As Table 1 shows, the value of α is 0.5 for OPT. This suggests that giving the same contribution to both labeled and unlabeled nodes produces a higher significant performance than both CO and GP when no initial prediction is available.

OPT-CO indicates the result when we use CO as the initial predictions for the unlabeled nodes. As it is shown in Table 1, this prediction decreases the F1 performance of OPT from 67.19% to 63.45%. This reduction is expected because the optimization framework has to optimize toward the polarity of both adjacent and d-nodes. Since CO produces poor performance in predicting the polarity of d-nodes, adding it to OPT reduces OPT’s performance.

Finally, OPT-GP gives the result when we use GP (see Equation 15) as the initial predictions for the unlabeled nodes. It outperforms both CO and GP by 21.16% and 2.78% in F1 score and the improvements are significant. OPT-GP also outperforms OPT by 1.86%. This result suggests that when we have better initial predictions, the performance of the optimization framework increases. Here the value of parameter α is set to 0.7 which emphasizes the important role of the labeled data (seeds) in the learning process. We study the effect of this parameter in the next Section.

6.2.1 Parameter Analysis

In this section we study the effect of parameter α on our optimization framework. As we mentioned before, this parameter has been tuned over 10% of the training data by a greed search in the [0.1, 1] range. We plot the F1 performance of different approaches (discussed in Table 1) on the test set with respect to parameter α . Note that in case of OPT the value of the d-nodes is 0 and therefore the parameter α has to be greater than 0, otherwise $f(x_i) = 0$ for $i=l+1 \dots n$ (See Equation 4).

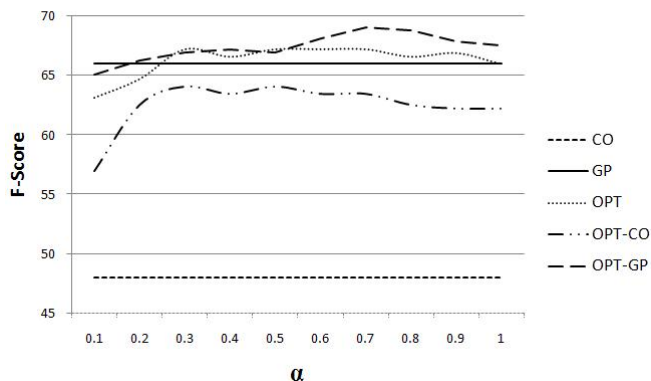


Figure 2. The Effect of α on Polarity Inference

The results are shown in Figure 2. As it is clear, the best performance is obtained when we use the optimization framework in conjunction with the GP predictions, OPT-GP, and the worst performance belongs to CO. In addition, both OPT and OPT-GP perform better than both baselines for $\alpha \geq 0.3$.

As expected, learning the predictions from GP, i.e. OPT-GP, improves the performance of OPT for all the values of α except when $\alpha = 0.3$ and $\alpha = 0.5$ where the performance of OPT is slightly higher than OPT-GP. This small reduction could be because of the noise in the unlabeled data. Figure 2 also shows that OPT and OPT-GP outperform OPT-CO independent from the value of α .

As expected, the smaller values of α produce lower performances. This shows that the labeled data play a crucial role in the learning process. However, Figure 2 shows that, to a lesser extent, learning from unlabeled data is also important. This is because when the optimization framework only learns from the labeled data, i.e. when $\alpha=1$, the performance of both OPT and OPT-GP decreases. This indicates the importance of the unlabeled data in the learning process.

6.3 Sentiment Classification Performance

The aim of SC is to assign a polarity label (positive or negative) to any given review. We expect that the performance of SC to be higher when we use an opinion lexicon with higher quality.

As the ground truth, we treated all the reviews with 1 or 2 stars as negative reviews, and the reviews with 4 or 5 stars as positive reviews. We obtained a total number of 14K reviews (balance on positive and negative classes) from the review dataset. To perform the SC experiments, we learned new opinion entities (SEs) from the cQA dataset and tested their SC performance on the 14K reviews.

We performed the word-matching-based SC as follows: given a review, the sentiment score of the review was computed as the sum of the polarity scores of the SEs that appear in the review. An overall positive sentiment score indicates a positive review; and negative otherwise. We also considered negations and disjunctive clauses as we explained in Section 4.1. Here we do not use any classifier in order to emphasize the quality of the lexicons. A higher performance can be obtained if we use an appropriate classifier.

We constructed separate polarity graphs for each value of N-Gram ($N=1, 2, 3$) and learned the *Top 1000* SEs that have sufficiently

Table 2. SC Performance on Positive Reviews

Lexicon	Precision	Recall	F1	imp
Seed_Lexicon	69.59	96.68	80.93	-
cQA_OPT	71.85	96.58	82.40*	+1.47
cQA_OPT-GP	72.34	96.59	82.72*	+1.79

Table 3. SC Performance on Negative Reviews

Lexicon	Precision	Recall	F1	imp
Seed_Lexicon	97.36	42.88	59.54	-
cQA_OPT	96.57	56.16	71.02*	+11.5
cQA_OPT-GP	96.45	58.80	73.06*	+13.5

Table 4. SC Performance Using on All the Reviews

Lexicon	Precision	Recall	F1	imp
Seed_Lexicon	76.28	69.78	72.89	-
cQA_OPT	79.32	76.37	77.82*	+4.93
cQA_OPT-GP	79.90	77.70	78.78*	+5.89

high confidence, i.e. $|f(x_i)| \geq 0.5$, for each set⁹. We then stored all these SEs into a lexicon to perform SC. Here, we only perform the experiments with the OPT and OPT-GP methods as they are the best performing methods based on the results of the previous Section. All the other parameters are set as reported in the previous Section, i.e. $\gamma = 0.5$ and $\alpha = 0.5$ for OPT, and $\gamma = 0.5$ and $\alpha = 0.7$ for OPT-GP.

Tables 2-4 show the performance of SC using different opinion lexicons and for different types of reviews (positive, negative, and all reviews respectively). The *Seed Lexicon* only contains the seeds, while the other lexicons, namely *cQA-OPT* and *cQA-OPT-GP*, contain the combination of seeds and SEs (mined from the cQA dataset) where OPT and OPT-GP were used for polarity inference respectively. The "imp" column shows the amount of F1 improvement over the *Seed Lexicon*.

As Tables 2 and 3 show *Seed Lexicon* produces a high F1 performance of 80.93% for the positive class, but a poor F1 performance of 59.54% for the negative class. We expected the *Seed Lexicon* to have high precision but low recall for SC. But this is only the case for the negative class.

To find the reason, we count the number of times that seeds occur in positive and negative reviews and it turns out that the positive seeds occur more frequently than negative ones. This affects the performance of our word-matching-based sentiment classifier. Table 5 shows the statistics. The "w negation" column means we take into account the negation words/rules as well, i.e. a negated positive (negative) seed increases the count of negative (positive) seeds. The "w/o negation" column reports the statistics without considering negation rules/words.

Table 5. Occurrences of Seeds in Reviews

	Pos Reviews		Neg Reviews	
	w negation	w/o negation	w negation	w/o negation
Pos Seeds	48,704	49,135	26,183	29,007
Neg Seeds	6,855	6,424	25,234	22,410
Pos/Neg Ratio	7.10	7.65	1.04	1.29

As Table 5 shows, the occurrence of positive seeds is much greater than the negative seeds in the positive reviews (7.10 and 7.65 times greater than with and without considering negation words/rules respectively)¹⁰. As such, the word-matching-based sentiment classifier is able to correctly label many of the positive reviews as positive. This justifies the high recall of *Seed Lexicon* for the positive class (96.68%). On the other hand, we found that the occurrence of positive seeds is slightly higher than the negative seeds in the negative reviews (1.04 and 1.29 times greater respectively). This seems to indicate that people tend not to use many negative words even in negative reviews. This causes some of the negative reviews to be wrongly labeled as positive by the word-matching-based sentiment classifier. This in turn results in the relatively low precision of the *Seed Lexicon* for the positive class (69.59%).

As shown in Table 5, the occurrence of positive seeds is slightly greater than the negative seeds in negative reviews (1.29 times greater). At the same time the occurrence of negation words/rules in negative reviews is greater than positive reviews. The above two indicators show that, in the negative reviews, users usually use negated positive seeds to express their negative opinions. This is consistent with the *positive encouragement principle* of critique, i.e. the shortcomings can be pointed out in a positive manner.

Table 2 and Table 3 show that the *cQA_OPT* and *cQA_OPT-GP* lexicons result in significant improvement over *Seed Lexicon* for both positive and negative classes, with a greater improvement on the negative class.

Table 4 shows the overall SC performance on all the reviews. The results show that both cQA lexicons significantly outperform the *Seed Lexicon*. Overall *cQA_OPT* results in 4.93% improvement over the *Seed Lexicon* (77.82% vs. 72.89%) while the *cQA_OPT-GP* result in 5.89% improvement over the *Seed Lexicon* (78.78% vs. 72.89%).

7. CONCLUSION

In this paper we focused on mining slang and urban opinion words and phrases from community-based question answering (cQA) archives. In the cQA context, such opinion entities can be used for discriminating opinion from factual questions, and answer summarization. The opinion entities are also useful for different tasks of sentiment analysis like review mining and sentiment classification. In this paper, we first utilized the opinion words with already known polarities (seeds) to extract a set of candidate opinion entities (or significant entities) from the community questions. We then formulated the polarity inference

⁹ We observed the same trend of performances with different values of K. Due to space limitation we omit the related curve.

¹⁰ The number of negation words is 4,777 and 13,040 in positive and negative reviews respectively.

task as a semi-supervised learning task in the graph context where the seeds and significant entities were modeled as the graph nodes. The graph connectivity information was then used to infer the polarity of significant entities. Our method is able to utilize both labeled and unlabeled data to learn the polarity of the entities and do not require dictionary-based relations (such as *synonym*, *antonym*, or *hyponym*) to construct the graph. We experimentally showed that our approach is effective in detecting new opinion entities and inferring their polarities. Furthermore, we showed that learning from both labeled and unlabeled data play a crucial role in inferring the polarity of candidate opinion entities.

In the future we aim to improve the performance of our approach and utilize the constructed lexicon for opinion question analysis in the context of community question services.

8. REFERENCES

- [1] Agichtein, E., Y. Liu, et al. (2009). "Modeling information-seeker satisfaction in community question answering." *ACM Trans. Knowl. Discov. Data* 3(2): 1-27.
- [2] Balahur, A., E. Boldrini, et al. (2009). Opinion and generic question answering systems: a performance analysis. In *Proceedings of the ACL-IJCNLP 2009*.
- [3] Balahur, A., E. Boldrini, et al. (2009). Towards the definition of requirements for mixed fact and opinion question answering systems. *Proceeding of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*. Hong Kong, China, ACM: 25-28.
- [4] Gamon, M. and A. Aue (2005). Automatic identification of sentiment vocabulary: exploiting low association with known sentiment terms. *Proceedings of the ACL Workshop on Feature Engineering for Machine Learning in Natural Language Processing*. Ann Arbor, Michigan, Association for Computational Linguistics: 57-64.
- [5] Ghahramani, X. Z. a. Z. (2002). Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon University.
- [6] Hassan, A. and D. Radev (2010). Identifying text polarity using random walks. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden, Association for Computational Linguistics: 395-403.
- [7] Hatzivassiloglou, V. and K. R. McKeown (1997). Predicting the semantic orientation of adjectives. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*. Madrid, Spain, Association for Computational Linguistics: 174-181.
- [8] Hu, M. and B. Liu (2004). Mining and summarizing customer reviews. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. Seattle, WA, USA, ACM: 168-177.
- [9] Iadh Ounis, M. d. R., Craig Macdonald, Gilad Mishne, and Ian Soboroff (2006). Overview of the TREC 2006 blog track. *Proceedings of the 15th Text Retrieval Conference (TREC)*.
- [10] Jaap Kamps, M. M., Robert J. Mokken, and Maarten de Rijke (2004). Using wordnet to measure semantic orientation of adjectives. *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004)*. IV: 1115-1118.
- [11] Kanayama, H. and T. Nasukawa (2006). Fully automatic lexicon expansion for domain-oriented sentiment analysis. *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Sydney, Australia, Association for Computational Linguistics: 355-363.
- [12] Kim, S.-M. and E. Hovy (2004). Determining the sentiment of opinions. *Proceedings of the 20th international conference on Computational Linguistics*. Geneva, Switzerland, Association for Computational Linguistics: 1367.
- [13] Li, B., Y. Liu, et al. (2008). Exploring question subjectivity prediction in community QA. *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. Singapore, Singapore, ACM: 735-736.
- [14] Li, F., Y. Tang, et al. (2009). Answering opinion questions with random walks on graphs. *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*. Suntec, Singapore, Association for Computational Linguistics: 737-745.
- [15] Liu, B. (2010). *Sentiment Analysis and Subjectivity*, Handbook of Natural Language Processing, Second Edition.
- [16] Luxburg, U. (2007). "A tutorial on spectral clustering." *Statistics and Computing* 17(4): 395-416.
- [17] Mohammad, S., C. Dunne, et al. (2009). Generating high-coverage semantic orientation lexicons from overtly marked words and a thesaurus. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2 - Volume 2*. Singapore, Association for Computational Linguistics: 599-608.
- [18] Nobuhiro Kaji, M. K. (2007). Building Lexicon for Sentiment Analysis from Massive Collection of HTML Documents *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.
- [19] Pang, B. and L. Lee (2008). "Opinion Mining and Sentiment Analysis." *Found. Trends Inf. Retr.* 2(1-2): 1-135.
- [20] Popescu, A.-M. and O. Etzioni (2005). Extracting product features and opinions from reviews. *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. Vancouver, British Columbia, Canada, Association for Computational Linguistics: 339-346.
- [21] Qiu, G., B. Liu, et al. (2009). Expanding domain sentiment lexicon through double propagation. *Proceedings of the 21st international joint conference on Artificial intelligence*. Pasadena, California, USA, Morgan Kaufmann Publishers Inc.: 1199-1204.
- [22] Rao, D. and D. Ravichandran (2009). Semi-supervised polarity lexicon induction. *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*. Athens, Greece, Association for Computational Linguistics: 675-682.
- [23] Sakai, T., D. Ishikawa, et al. (2011). Using graded-relevance metrics for evaluating community QA answer selection. *Proceedings of the fourth ACM international conference on*

Web search and data mining. Hong Kong, China, ACM: 187-196.

- [24] Sebastiani, A. E. a. F. (2006). SentiWordNet: A publicly available lexical resource for opinion mining. P. o. L. R. a. E. (LREC).
- [25] Stone, P. J. and E. B. Hunt (1963). A computer approach to content analysis: studies using the General Inquirer system. Proceedings of the May 21-23, 1963, spring joint computer conference. Detroit, Michigan, ACM: 241-256.
- [26] Takamura, H., T. Inui, et al. (2005). Extracting semantic orientations of words using spin model. Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Ann Arbor, Michigan, Association for Computational Linguistics: 133-140.
- [27] Tomasoni, M. and M. Huang (2010). Metadata-aware measures for answer summarization in community Question Answering. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. Uppsala, Sweden, Association for Computational Linguistics: 760-769.
- [28] Turney, P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Philadelphia, Pennsylvania, Association for Computational Linguistics: 417-424.
- [29] Turney, P. D. and M. L. Littman (2003). "Measuring praise and criticism: Inference of semantic orientation from association." ACM Trans. Inf. Syst. 21(4): 315-346.
- [30] Velikovich, L., S. Blair-Goldensohn, et al. (2010). The viability of web-derived polarity lexicons. Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Los Angeles, California, Association for Computational Linguistics: 777-785.
- [31] Wang, F. and C. Zhang (2006). Label propagation through linear neighborhoods. Proceedings of the 23rd international conference on Machine learning. Pittsburgh, Pennsylvania, ACM: 985-992.
- [32] Wang, K., Z. Ming, et al. (2009). A syntactic tree matching approach to finding similar questions in community-based qa services. Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval. Boston, MA, USA, ACM: 187-194.
- [33] Wilson, T., J. Wiebe, et al. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing. Vancouver, British Columbia, Canada, Association for Computational Linguistics: 347-354.
- [34] Zhu, X., Ghahramani, Z., & Lafferty, J. (2003). Semi-supervised learning using Gaussian fields and harmonic functions. The 20th International Conference on Machine Learning (ICML).
- [35] Islam, A. and D. Inkpen (2008). "Semantic text similarity using corpus-based word similarity and string similarity." ACM Trans. Knowl. Discov. Data 2(2): 1-25.

APPENDIX

A. DERIVATION FOR THE ENERGY FUNCTION $E(\mathbf{F})$

Since $f(x_i) = y_i \forall i = 1 \dots l$, clearly:

$$\gamma(\mathbf{f} - \mathbf{y})^T \mathbf{I} (\mathbf{f} - \mathbf{y}) = \gamma \sum_{i=l+1}^n (f_i - y_i)^2$$

Now we need to show that:

$$\mathbf{f}^T \mathbf{\Omega} \mathbf{f} = (1 - \gamma) \sum_{i=l+1}^n \left(\sum_{j=1}^l \alpha w_{ij} (f_i - f_j)^2 + \sum_{j=l+1}^n (1 - \alpha) w_{ij} (f_i - f_j)^2 \right)$$

By definition $\mathbf{\Omega} = \mathbf{D} - \mathbf{T}$, so we have:

$$\begin{aligned} \mathbf{f}^T \mathbf{\Omega} \mathbf{f} &= \mathbf{f}^T \mathbf{D} \mathbf{f} - \mathbf{f}^T \mathbf{T} \mathbf{f} = \sum_{i=1}^n d_{ii} f_i^2 - \sum_{i=1}^n \sum_{j=1}^n T_{ij} f_i f_j = \\ &= \frac{1}{2} \left(\sum_{i=1}^n d_{ii} f_i^2 - 2 \sum_{i=1}^n \sum_{j=1}^n T_{ij} f_i f_j + \sum_{j=1}^n d_{jj} f_j^2 \right) \\ &= \frac{1}{2} \left(\sum_{i=1}^n \sum_{j=1}^n T_{ij} (f_i - f_j)^2 \right) \end{aligned}$$

Considering the symmetric matrix T defined in Equation (6), we have:

$$\begin{aligned} \mathbf{f}^T \mathbf{\Omega} \mathbf{f} &= \frac{1}{2} \left(\sum_{i=1}^l \sum_{j=l+1}^n T_{ij} (f_i - f_j)^2 \right) + \\ &= \frac{1}{2} \left(\sum_{i=l+1}^n \sum_{j=1}^l T_{ij} (f_i - f_j)^2 \right) + \\ &= \frac{1}{2} \left(\sum_{i=l+1}^n \sum_{j=l+1}^n T_{ij} (f_i - f_j)^2 \right) \\ &= \sum_{i=l+1}^n \sum_{j=1}^l T_{ij} (f_i - f_j)^2 + \\ &= \frac{1}{2} \left(\sum_{i=l+1}^n \sum_{j=l+1}^n T_{ij} (f_i - f_j)^2 \right) \\ &= \sum_{i=l+1}^n \sum_{j=1}^l \alpha (1 - \gamma) w_{ij} (f_i - f_j)^2 + \\ &= \frac{1}{2} \left(\sum_{i=l+1}^n \sum_{j=l+1}^n 2(1 - \alpha)(1 - \gamma) w_{ij} (f_i - f_j)^2 \right) \\ &= (1 - \gamma) \sum_{i=l+1}^n \left(\sum_{j=1}^l \alpha w_{ij} (f_i - f_j)^2 + \sum_{j=l+1}^n (1 - \alpha) w_{ij} (f_i - f_j)^2 \right) \blacksquare \end{aligned}$$