Learning Subjective Language: Feature Engineered vs. Deep Models

Muhammad Abdul-Mageed
Natural Language Processing Lab
University of British Columbia
muhammad.mageeed@ubc.ca

Abstract

Treatment of subjective language is a vital component of a sentiment analysis system. However, detection of subjectivity (i.e., subjective vs. objective content) has attracted far less attention than sentiment recognition (i.s., positive vs. negative language). Particularly, online social context and the structural attributes of communication therein promise to help improve learning of subjective language. In this work, we describe successful models exploiting a rich and comprehensive feature set based on the structural and social context of the Twitter domain. In light of the recent successes of deep learning models, we also effectively experiment with deep gated recurrent neural networks (GRU) on the task. Our models exploiting structure and social context with an SVM achieve $> 12\%$ accuracy higher than a competitive baseline on a blind test set. Our GRU model yields even better performance, reaching 77.19 (i.e., $\sim 14.50\%$ higher than the baseline on the same test set, $p < 0.001$).

1. Introduction

Ability to detect subjective language (i.e., aspects of language expressing opinions, feelings, evaluations, and speculations (Banfield, 1982)) is an important part of any real-world sentiment system where a unit of analysis is usually labeled as either objective (e.g., *I read the book.*) or subjective. Subjective texts are further classified into sentiment categories as *positive* (e.g., *This market is spectacular!*), *negative* (e.g., *This machine is unfortunately very slow.*), or *mixed* (e.g., *The new models are powerful, but quite memory-intensive!*). In spite of an excellent (early) thread of literature targeting learning subjective language that focused on utilizing lexical and syntactic information (Wiebe et al., 2004; Wilson et al., 2006)), gender (Burger et al., 2011; Rao et al., 2010; Volkova et al., 2013; Volkova et al., 2015), and discourse features (e.g., punctuation, emoticons) (Benamara et al., 2011), the field has focused more on sentiment or polarity classification rather than subjectivity.

Particularly social media communication takes place in a very different, yet rich, context: First, Twitter language diverges from the 'standard' offline language in various *structural* ways. For example, Twitter tweets are a maximum of 140 characters per tweet. Twitter is also an environment where users re-tweet other users, address them using an '@' sign, tag tweets and/or launch tweet campaigns using hashtags, share URLs, etc. Rather than viewing these unique structural attributes of the Twitter domain as challenges, we seek to exploit them for learning subjectivity in the context of the microblogging platform.

Second, communication on Twitter happens against its wider *social* context where user identities, gender, race, age, economic class, etc. are all attributes that afford cues which can be leveraged for social meaning extraction tasks like that of subjectivity. Although (at times scattered) features based on the structure of Twitter language and its social context have been used in the literature, a unified and systematic analysis of the collection of *structural and social context* features that can inspire further work in the field, especially for the Arabic language, is lacking. As such, we describe novel and successful models exploiting a rich feature set (totaling 30 features thematically organized in 11 feature groups) based on the structural and social attributes of the the Twitter domain. Examples of *structural* features we employ include use of hashtags, non-standard typography (e.g., letter repetition, use of emoticons), and use of URLs. Instances of *social* features we leverage include user id and user gender. We provide a more detailed account of our feature set in Section 4.1..

Third, while there are several methods for feature selection, including for text classification (e.g., (Dash and Liu, 1997; Yang and Pedersen, 1997; Forman, 2003; Chandrashekar and Sahin, 2014)), finding the relevant features and the best combinations of these from a feature set composed of a large number of features can be challenging, if not impossible. In this work, we introduce two methods of feature selection aimed at identifying the best performing feature combinations from among the 30 proposed features.

Finally, deep learning of natural language (LeCun et al., 2015; Goodfellow et al., 2016; Goldberg, 2016) has shown impressive successes in recent years. It is yet unknown, however, to what extent a deep learning system would compare to a system based on careful feature engineering using domain knowledge of the type provided in this work in the context of subjectivity classification. Our work here seeks to at least partially bridge this gap by comparing a feature-engineered system to a carefully-designed deep learning system tackling the problem.

Overall, we make the following contributions: (1) We propose a rich set of structural and social context features that we exploit for learning subjective language onnline (i.e., on a Twitter dataset), (2) We describe two feature selection methods that enable a semi-exhaustive search for the best feature combinations from a large number of features that are otherwise hard to search, and (3) We develop a highly effective gated recurrent neural network model for the task, showing the utility of this class of methods and how it is that these compare to our expertly hand-crafted system exploiting the features we introduce.

The rest of the paper is organized as follows: In section 2., we discuss related work. In Section 3., we describe our dataset. In Section 4., we describe our models with hand-crafted features. In Section 6., we introduce our model based on gated recurrent neural networks and present its results acquired with it. In Section, 7. we conclude.

2. Related Work

Subjectivity in Natural Language *Subjectivity* in human language, as introduced earlier, refers to aspects that express opinions, feelings, evaluations, and speculations (Banfield, 1982). There is a vast literature on subjectivity and sentiment analysis. Early computational treatment of subjectivity (e.g., (Wiebe, 2000; Wilson et al., 2006)) focused on the lexical and syntactic cues characterizing subjective texts. Our work differs in that we utilize structural and social context features. More recent works investigate exploiting demographic features of the type we incorporate in our feature set here. Especially gender has received significant attention as an attribute that correlates with subjective language (Burger et al., 2011; Rao et al., 2010; Volkova et al., 2013; Volkova et al., 2015). Discourse features, including punctuation- and emoticon-based features, have also been studied in the context of improving subjectivity detection (Benamara et al., 2011).

A number of social context features have also been used for predicting phenomena related to subjectivity. For example, (Persing and Ng, 2014) employ information related to political orientation, relationship status, and health behavior (e.g., drinking, smoking) to predict voting from comments posted on a polling social platform. Similarly, (Thomas et al., 2006) report benefiting from user mentions (e.g., using the "@") network for predicting votes and (Tan et al., 2011) acquire enhanced sentiment classification by incorporating the Twitter follower/followee and user mentions network. (Hasan and Ng, 2013) incorporate sequential user interactions and ideological orientation in debate web fora for stance detection. (Deng et al., 2014) similarly use network-based information between users to improve sentiment classification both at the post and user levels. (Ren et al., 2016) embed user tweets and topics in a neural framework for improving Twitter sentiment analysis. Likewise, a number of researchers, e.g., (Mohammad and Kiritchenko, 2015; Purver and Battersby, 2012; Wang et al., 2012) makes use of Twitter hashtags as a way to automatically label data for the related task of emotion detection, while a string of works considers textual clues (e.g., negation, epistemic modality) interacting with subjective language (Wiegand et al., 2010; Kennedy and Inkpen, 2006). Our work is is similar in that we exploit a wide range of these features, while expanding them and proposing methods enabling searching for their best combinations in the context of classification.

For modeling the related task of sentiment, researchers have typically exploited lexical features using simple frequency statistics of input text (Wiebe, 2000; Wiebe et al., 2004), or modeling the semantics of certain word categories, e.g., dynamic and gradable adjectives (Hatzivassiloglou and Wiebe, 2000) or different semantic classes of verbs (Benamara et al., 2007; Breck et al., 2007)).

A considerable body of the literature has focused on developing or learning polarity lexica (Lin and Hauptmann, 2006; Baccianella et al., 2010; Turney, 2002). Other works have exploited syntactic features like part of speech tags (Gamon, 2004; Hatzivassiloglou and McKeown, 1997) and different *N*-gram windows as a measure to capture (potentially syntactic) context beyond single words (Ng et al., 2006; Cui et al., 2006), syntactic constituents, e.g., (Klenner et al., 2009; Wilson et al., 2005), dependency parses, e.g., (Kessler and Nicolov, 2009; Zhuang et al., 2006; Ng et al., 2006), etc. A few studies have focused on languages of rich morphology, including (Abdul-Mageed et al., 2014) who built systems using gold-processed, treebank data exploiting mophosyntactic information. Other works on Arabic involved building resources that were then used for developing models primarily based on N-gram features (Aly and Atiya, 2013; ElSahar and El-Beltagy, 2015; Mourad and Darwish, 2013) or sub-word information (Abdul-Mageed, 2017b; Abdul-Mageed, 2017a). Some works have focused on modeling dialects (Abdul-Mageed et al., 2014), or the related task of emotion (Abdul-Mageed et al., 2016), yet these remain relatively limited. Recent efforts to collect large-scale Arabic dialectal corpora promise to aid dialect-specific sentiment research (Abdul-Mageed et al., 2018). The focus of our work is different in that we target structural and social features.

Deep Learning Models An increasingly growing number of studies have applied deep neural networks to the problem of sentiment analysis. These include, e.g., (Labutov and Lipson, 2013; Maas et al., 2011; Tang et al., 2014b; Tang et al., 2014a) who learn sentiment-specific word embeddings (Bengio et al., 2003; Mikolov et al., 2013) from neighboring text. Some studies have focused on learning semantic composition (Mitchell and Lapata, 2010; Socher et al., 2013; Irsoy and Cardie, 2014; Li et al., 2015; Le and Mikolov, 2014; Tang et al., 2015) for modeling sentiment. Long-short term memory (LSTM) (Hochreiter and Schmidhuber, 1997) and Gated Recurrent Neural Nets (GRUs) (Cho et al., 2014; Chung et al., 2015), variations of recurrent neural networks (RNNs) have also been used successfully for sentiment analysis (Ren et al., 2016; Liu et al., 2015; Tai et al., 2015; Tang et al., 2015; Zhang et al., 2016). Convolutional neural networks (CNNs) have also been quite successful, including on sentiment analysis (Blunsom et al., 2014; Kim, 2014; Zhang et al., 2015). A review of neural network methods for NLP can be found in (Goldberg, 2016). Our work is similar to these works in that we use GRUs for learning subjective language, basically a text classification task.

For a more comprehensive background on modeling subjectivity and sentiment, readers can refer to a number of excellent comprehensive overviews, including (Pang and Lee, 2008), (Liu, 2012), and (Montoyo et al., 2012). In addition, (Benamara et al., 2017) provide a more recent thorough review of various aspects of evaluative text, including some aspects of social context [1] (e.g., social network structure and user profiles).

[1](Benamara et al., 2017) use the term 'extra-linguistic information' to refer to what we call *social context* in this paper.

3. Data Set and Annotation

For this work, we collect a corpus of $3,015$ Arabic Tweets from the public Twitter timeline and task two college-educated native speakers of Arabic on labeling the data after providing detailed annotation instructions and training as described in (Abdul-Mageed et al., 2014). The data were manually inspected for possible duplicates before we shared with the annotators, and so the $3,015$ are unique. Table 1 shows the the distribution of the SSA categories over the data. As Table 1 shows, 47.36% of the data are assigned an objective (OBJ) tag and the remaining 52.64% has one of the various subjective tags: Positive (S-POS), negative (S-NEG), and mixed (S-MIXED). [2] Inter-annotator agreement on the data reached a Cohen (Cohen, 1996) Kappa *(K)*$= 85\%$. We take the labels assigned by a random judge from among the two annotators to be our gold standard. As Arabic is known to have multiple dialects in addition to its modern standard variety, we also ask annotators to assign each tweet a tag representing whether the variety is Modern Standard Arabic (MSA) or dialectal Arabic (DA). The MSA of the corpus comprises $1,466$ ($\% = 48.62$) tweets, and the dialectal part comprises $1,549$ ($\% = 51.38$) tweets. We do not exploit these language variety tags for this work.

4. Models Based on Hand-Crafted Features with Support Vector Machines

We both use a classical machine learning classifier, introduced here, and a deep learning classifier, which we will introduce in Section 6.. For our models with hand-crafted features, we use an SVM (Vapnik, 1995) classifier with a linear kernel. SVMs are know to perform well on text classification (Joachims, 2002), especially with carefully-designed feature sets. We now turn to introducing our gated recurrent neural network model.

4.1. Features

In order to exploit the structural and social context, we introduce a very rich feature set composed of a total of 30 features. To facilitate reference, we divide these features thematically (with as much coherence per group as is possible) into 11 groups. Although the features target the Twitter domain, we believe they can also be exploited for other domains like chat fora. Our features are inspired by research within the sentiment literature, but also by related areas such as stance detection (Hasan and Ng, 2013), voting prediction (Thomas et al., 2006; Persing and Ng, 2014), and social media and computer-mediated communication (Herring, 2007; Androutsopoulos and Beißwenger, 2008; Herring et al., 2013; Bieswanger, 2013). We now describe our feature set.

User Gender: Inspired by gender variation research exploiting social media data, e.g., (Herring, 1994), and previous research on sentiment analysis (Volkova et al., 2013), we applied three gender (gen) features corresponding to the set {*hasMale, hasFemale, unknown*}. (Abdul-Mageed et al., 2014) suggest that there is a relationship between politeness strategies and sentiment expression. And gender variation research in social media has found that expression of linguistic politeness (cf. (Brown and Levinson, 1987)) differs based on the gender of the user: (Herring, 1994) identified gender differences in expressions of linguistic politeness in ways that interact with sentiment expression. (Herring, 1994) maintains that women are more likely than men to observe positive politeness through, e.g., thanking, while men prefer 'candor' and assertion of opinion, even when it conflicts with other people's opinions; such behaviors might interact with the type of subjectivity data carries.

User ID: The user id (uid) labels are inspired by research on Arabic Twitter [citation removed for blind review] showing that a considerable share of tweets is produced by organizations such as news agencies as opposed to lay users. Hence, two features from the set {*person, organization*} are employed for classification. The assumption we make is that tweets from persons will have a higher correlation with expression of subjectivity than those from organizations.

URL and Quotation: *(a). hasURL:* A binary feature indicating the existence of a URL in a tweet or lack thereof. *(b). hasQuotation:* A binary feature indicating whether a unit of analysis has quotation marks or not.

Existence of Latin: *hasLatin:* A binary feature indicating the existence of a Latin-alphabeted word in a tweet or lack thereof.

Speech-like Features: *(a). hasLetterRepetition:* A binary feature indicating the existence of a sequence of the same letter within a given word with a frequency > 3 in a tweet or lack thereof. *(b). hasLaughter:* A binary feature indicating the existence of the laughter word 'haha' or the laughter word 'hehe' in a tweet or lack thereof.

Emoticons: *(a). hasEmoticon:* A binary feature indicating the existence of an emoticon from the set { *';)', ':)', ':(', ';(', ':d', '(:', '):', ':D'*} in a tweet or lack thereof. *(b). hasPositiveEmoticon:* A binary feature indicating the existence of an emoticon from the set {*':)', ':d', ':; '(:', ':D'*} in a tweet or lack thereof. *(c). hasNegativeEmoticon:* A binary feature indicating the existence of an emoticon or emoticon-like interjection from the set {*':(', '):', 'ugh'*} in a unit of analysis or lack thereof.

Hashtags and Retweets: *(a). hasHashtag:* A binary feature indicating the existence of a hashtag '#' in a data point or lack thereof. *(b). hasMultipleHashtags:* A binary feature indicating the existence of two or more hashtags in a tweet or lack thereof. *(c). hasLongHashtag:* A binary feature indicating the existence of a hashtag of either length > 9 characters or with an underscore '_' in a data point or lack thereof. *(d). isRetweet:* A binary feature indicating whether a post is a retweet (has the prefix 'RE,' as is the norm in Twitter usage) or not.

Addressees: *(a). hasAddressee:* A binary feature indicating the existence of a username (as detected by the existence of an initial '@' sign in a string) in a tweet or lack thereof. *(b). hasMultipleAddressees:* A binary feature indicating the existence of two or more usernames in a tweet or lack thereof.

Punctuation: *(a). hasExclamation:* A binary feature indi-

[2] Although the focus of the current work is on the binary classification task of detecting whether a given tweet is OBJ or SUBJ, we also provide negative experiments on the sentiment data (as described in Section 5.).

Table 1: Data statistics

Data set	OBJ	S-POS	S-NEG	S-MIXED	# Tweets
MSA	960 (65.48%)	226 (15.42%)	186 (12.69%)	94 (6.41%)	1,466
DA	468 (30.21%)	257 (16.59%)	573 (36.99%)	251 (16.20%)	1,549
ALL	1,428 (47.36%)	483 (16.02%)	759 (25.17%)	345 (11.44%)	3,015

cating the existence of an exclamation mark in a data point or lack thereof. *(b). hasMultipleExclamation:* A binary feature indicating the existence of two or more exclamation marks in a tweet or lack thereof. *(c). hasQuestionMark:* A binary feature indicating the existence of a question mark in a unit of analysis or lack thereof. *(d). hasMultipleQuestionMark:* A binary feature indicating the existence of two or more question marks in a tweet or lack thereof.

Word Length: *(a). hasAvgShortWords:* A binary feature indicating whether the average word length of a unit of analysis is < 5 characters or not. *(b). hasAvgMediumWords:* A binary feature indicating whether the average word length of a tweet is at least 5 characters but < 7 characters or not. *(c). hasAvgLongWords:* A binary feature indicating whether the average word length of a tweet is > 7 characters or not.

Unit Length: *(a). hasShortLength:* A binary feature indicating whether the length of a unit of analysis is < 4 words or not. *(b). hasMediumLength:* A binary feature indicating whether the length of a unit of analysis is at least 4 words but < 8 words or not. *(c). hasLongLength:* A binary feature indicating whether the length of a tweet is at least 8 words but < 14 words or not. *(d). hasVeryLongLength:* A binary feature indicating whether the length of a data point is > 13 words or not.

4.2. Experimental Setup

Data Splits & Settings: We split the data into 80% training (TRAIN), 10% development (DEV), and 10% testing (TEST). We use three experimental settings as follows:

Individual Features (IVF): We add each of the individual features independently to the baseline bag-of-word (bow) setting and perform classification, thus measuring the utility of each of these features as combined with the simple bow baseline.

Whole Feature Set (WH): The whole feature set of 30 features is added to the baseline bow setting, and classification is performed. The way this setting is engineered is that any of the features that exist in any of the sentences used is added to the sentence vector, at the sentence level. This method allows identifying the utility of adding all the features combined on the classification task.

Feature Selection: Since some of the features may be more relevant than others to the task and since a feature can possibly perform differently based on the group of features it is used with, we also perform feature selection with a number of configurations, as follows:

Exhaustive feature group selection (FG): A search with all possible combinations of feature groups of the feature set is performed. In this setting, each group of the feature groups we described above is combined with zero or more groups, such that all possible combinations of the feature groups are considered. This method is better than the popular 'hill climbing' methods, whether in a forward or backward selection fashion. In *forward selection*, a given feature is added to a basic feature set, and if found useful, the feature is added to the basic feature set. Otherwise it is discarded, and the rest of the features are added in the same way to the basic feature set (which, after each iteration, includes more of the features of interest). The process continues until all features are considered, then the final performance is reported. Forward selection is described as 'hill climbing' search since it proceeds based on the potential gain each considered feature achieves in the classification process.

Backward selection is similar to forward selection, except that the classification starts with all the basic features, as well as all the features of interest, and a feature is dropped during each iteration to identify whether this ablation helps or hurts classification. The feature of interest is removed if its removal helps the classification, and the process is repeated. Like forward selection, backward selection proceeds based on potential gains removal of individual features can achieve. Exhaustive feature group search (FG) is better than hill climbing on feature groups in that during it, all possible combinations of groups of features are considered; hence any gains possible by any of such combinations are identified. This is different from hill climbing on feature groups, since hill climbing is not exhaustive and hence can miss possible feature group combinations that can achieve optimal performance. The down side of exhaustive search is its computational cost. However, this disadvantage is minor, since the process is performed offline. In addition, exhaustive search is practically possible only on a small feature set as the feature groups comprise here.

Monte Carlo feature selection (MC): A random sample of varying sizes from 1 to 30 of the individual features is added to the baseline bag-of-words setting, and classification is performed. This procedure is repeated $10K$ times, each time with a different random sample of a different size, such that different combinations of the individual features are possible. The Monte Carlo method is useful since, with a large number of iterations as in the case of $10K$, it is very likely that all possible combinations of individual features will be considered. The Monte Carlo method is preferred for mimicking exhaustive feature search with the individual 30 features. Attempting to perform individual feature exhaustive search with a procedure other than the Monte Carlo method would be extremely computationally costly and probably not needed, since processing the 30 social context features would mean $30!$ operations.

Procedure: We typically train classifiers on TRAIN, tune performance on DEV (e.g., to identify the performance of different sets of feature combinations and select overall best-performing feature set), and blind-test on TEST. For all the experiments, we use an SVM classifier with a linear kernel. We provide results on both DEV and TEST, as

appropriate.

Evaluation: Results are reported in terms of overall accuracy (acc) and F_1-score for the OBJ class (F_1-O) and the SUBJ class (F_1-S). Since the majority class in our training data is low ($= 52.64\%$), we use a baseline that is 10% higher. More specifically, we use performance with bag-of-words input (bow) on DEV (acc $= 62.67\%$) as our baseline.

4.3. Results

As Table 2 shows, on DEV, the whole feature set (WH) achieves a gain of 6.34% accuracy (acc) over the baseline bag-of-words (bow). In addition, the bow baseline is outperformed by the exhaustive group feature selection (FG) with 7.00% acc and by the Monte Carlo exhaustive feature selection method (MC) with 7.33% acc. Similarly, on TEST, the baseline is outperformed by WH and FG (with 12.74% acc for both cases), and by MC with 12.08% acc. All the gains are statistically significant ($p < 0.001$). Observably, TEST seems an easier set than DEV as indicated by its bow performance (at 66.23 acc) (vs. the baseline DEV bow, with acc $= 62.67$). Compared to the TEST bow performance, the models across all the experimental conditions on TEST are also highly successful and remain within statistical significance ($p < 0.001$): WH gains 9.18% acc, FG gains 9.18% acc, and MC gains 8.52% acc. We now turn to analyzing performance with each of our experimental settings.

FG Method: The FG method helps achieve the improvement with a number of feature group combinations. A consideration of these combinations shows that almost all the groups were chosen in one or another of them. In some of the selected combinations, some of the groups that were useful in other combinations were absent. For example, one of the combinations includes all the feature groups except the *hasLatin* feature, the *speech-like* features, and the *hashtag* features. These three specific feature groups were useful for the classification in other combinations that were also found to render the same classification improvement. This suggests that the FG feature selection method found intricate interactions among the groups. The importance of these groups of features is also reflected in the fact that the individual features within these groups were also selected via the MC method, whose performance we now turn to explaining.

MC Method: Similarly, in the MC method, several feature combinations were chosen. Again, an examination of these combinations shows that almost all the individual features were selected in one or another of the different combinations. For example, one of the combinations that achieved the best performance reported includes all the features except the three features *hasHashtag, hasQuestionMark,* and *hasIsMediumLength*.

IVF Method: Regarding experiments with the IVF feature engineering method, results show that the *gender (gen)* feature group, the *user ID (uid)* feature group, the *hasHashtag* feature, and the *hasURL* feature were useful for classification when added independently, as shown in Table 3.

Gender: The gender-based features proved useful for classification. In TRAIN, the distribution difference especially between the *female* and the *unknown* features within the objective and subjective classes is large enough to help classification: The *female* feature occurs in 25.14% of the subjective class data and 16.09% in the objective class data. For *unknown*, it occurs in 33.53% of the objective class cases and 14.92% of the subjective class. The *uid* group was also especially useful, with noticeably different distribution in TRAIN: The *person* feature occurs in 95.47% of the subjective class and in 80.29% of the objective class, whereas *organization* occurs in 19.71% in the objective class and 4.53% in the subjective class. A consideration of both TRAIN and DEV data shows that organizations seem to be more concerned with tweeting information objectively, perhaps as a way to gain credibility. After all, many of these organizations are news outlets interested in keeping their audiences' interest and trust, and (at least ostensibly,) unbiased coverage is important for them (Abdul-Mageed and Herring, 2008).

URL: The *hasURL* feature was also useful for classification. Based on TRAIN, tweets containing URLs are twice as likely to be in the objective (52.91%) class than the subjective class (23.17%). This is the case because URLs are more likely to be associated with information provision in the context of advertising, e.g., where users are encouraged to visit a website promoting some commodity. The following are two examples:

- (1) مهندس مدنى وارغب فى السفر للعمل فى السعودية http://bit.ly/iklvJh.

  Buck. 'mhnds mdnY wArgb fY Alsfr llEml fY AlsEwdyp http://bit.ly/iklvJh'

  Eng. '[I'm a] civil engineer and need a job in KSA http://bit.ly/iklvJh.'

- (2) برنامج يرنامج مشغلات المالتيميديا ٣٣١٠٥٠٢٠٣ ٦ فى أخر إصداراته http://goo.gl/fb/mBc2b.

  Buck. 'brnAmj yrnAmj m$glAt AlmAltymydyA 3.2.5.1306 fY Oxr ISdArAt http://goo.gl/fb/mBc2b.'

  Eng. 'Software software [sic] for playing multimedia 3.2.5.1306 in its latest release http://goo.gl/fb/mBc2b.'

Questions: Similarly, based on TRAIN, questions are more likely to occur in objective, information seeking tweets (7.83%) than in subjective tweets (6.84%). The following is an example of an objective question:

- (3) ممكن اعرف مين فى حركة ٦ ابريل من عين شمس عشان محتاج اتعاون معاهم.

  (**Buck.** 'mmkn AErf myn fY Hrkp 6 Abryl mn Eyn msEAn mHtAj AtEAwn mEAhm.'; **Eng.** 'Can I know who in April 6 Movement is in Ein Shams so that I contact them[?]').

Exclamation Marks: Unlike question marks, exclamation marks are quite expectedly more frequent in subjective cases than in objective cases in TRAIN. The *hasExclamation* feature occurred with a frequency of 6.84% in the subjective class and 3.12% in the objective class. Likewise,

Table 2: Results with whole set (WH), exhaustive group selection (FG), and Monte Carlo selection (MC)

				OBJ			SUBJ		
	setting	**acc**	**avg-f**	**prec**	**rec**	**f**	**prec**	**rec**	**f**
DEV	**base (bow)**	62.67	62.56	51.19	74.14	60.56	77.27	55.43	64.56
	WH	**69.00**	68.08	58.65	67.24	62.65	77.25	70.11	73.50
	FG	**69.67**	68.91	59.12	69.83	64.03	78.53	69.57	73.78
	MC	**70.00**	69.28	59.42	70.69	64.57	79.01	69.57	73.99
TEST	**bow**	66.23	65.69	54.67	70.09	61.42	77.42	63.83	69.97
	WH	**75.41**	73.96	68.10	67.52	67.81	79.89	80.32	80.11
	FG	**75.41**	73.96	68.10	67.52	67.81	79.89	80.32	80.11
	MC	**74.75**	73.27	67.24	66.67	66.95	79.37	79.79	79.58

Table 3: Individual features acquiring classification gains

					OBJ			SUBJ		
			Acc	**Avg-F**	**Prec**	**Rec**	**F**	**Prec**	**Rec**	**F**
DEV		**bow**	62.67	62.56	51.19	74.14	60.56	77.27	55.43	64.56
	gen	**+**	63.67	63.37	52.23	70.69	60.07	76.22	59.24	66.67
	uid	**+**	65.00	64.72	53.5	72.41	61.54	77.62	60.33	67.89
	hasHashtag	**+**	63.33	63.2	51.81	74.14	60.99	77.61	56.52	65.41
	hasPositiveEmot	**+**	63.00	62.88	51.5	74.14	60.78	77.44	55.98	64.98
	hasURL	**+**	68.00	67.22	57.25	68.1	62.2	77.16	67.93	72.25
TEST	**gen**	**+**	66.89	66.12	55.63	67.52	61.00	76.69	66.49	71.23
	uid	**+**	68.20	66.91	57.81	63.25	60.41	75.71	71.28	73.42
	hasHashtag	**+**	66.89	66.36	55.33	70.94	62.17	78.06	64.36	70.55
	hasPositiveEmot	**+**	66.23	65.69	54.67	70.09	61.42	77.42	63.83	69.97
	hasURL	**+**	71.15	69.59	62.18	63.25	62.71	76.88	76.06	76.47

the *hasExclamationRepetition* feature was more frequent in the subjective class (with 1.57%) than in the objective class (0.67%). The following is an example of a subjective tweet employing multiple exclamation marks:

- (4) أنا بتفرج على فيديو دلوقتى هيجيبلى كوابيس !!! واضح إن فى منافس لأحمد زبايدر !!! مش قادر.

  Buck. 'OnA btfrj ElY fydyw dlwqtY hyjyblY kwAbys !!! wADH In fY mnAfs lOHmd zbAydr !!! m$ qAdr.'

  Eng. 'I'm watching a video right now[.] I'll have nightmares!!! Clearly, there's a competitor to Ahmad Spider!!! I can't take it.'

Emoticons: Although emoticons are usually viewed as symbols associated exclusively to subjective language, our annotators indeed assigned OBJ tags to a number of cases where positive emoticons occur. Positive emoticons, however, were more frequent in the subjective class (2.97%) than in objective class (1.18%). The following is an example of a smiley face in a subjective tweet:

- (5) شكرا وإذا فى أى ملاحظة ترى أنا إتقبل النقد .. :)

  Buck. '$krA wI*A fY OY mlAHZp trY OnA Itqbl Alnqd ..'

  Eng. 'Thanks[!] And let me know if you have any feedback[;] I take criticism.. :)'

Hashtags: The *hasHashtag* feature is also useful for classification as, in TRAIN, the feature is more frequent in the subjective class (with 0.49%) than in the objective class (with 0.34%). In DEV, the feature only occurred in subjective cases. Since hashtags are sometimes used to mark the topic of a tweet and have the potential to contribute to the popularity of a (trending) topic, they are used for campaigning in Twitter. Indeed, hashtags have played an important role in online activism in the Arab world (and elsewhere). In TRAIN, it is clear that political campaigning is an important function for which hashtags are used.

Sometimes users employ hashtags of more than one word, where the words are simply concatenated (potentially separated by an underscore). Longer hashtags are especially more frequent in the TRAIN subjective class (with a frequency of 0.08%) than in objective class (where they are totally absent). The same bias occurs in DEV (where their frequency is 0.54% in the subjective class and zero in the objective class). Examples of such hashtags are #egygilrs, #mubarakregrets, #wheniwasakidithought, #newegypt, and #3eshnaooshifna (Eng. 'Look what is happening').[3]

[3]In May 2010, when the dataset was being collected, hashtags in Arabic Twitter were exclusively in English, as Twitter did not allow use of hashtags with Arabic words. As explained earlier, even if a user wanted to use a hashtag with an Arabic word, the word would not be clickable. As of the writing of this paper, Arabic Twitter users employ a mixture of Arabic and English hashtags.

5. Negative Experiments

In order to test the performance of the feature set we propose here on sentiment classification, we run experiments with all the three settings on the polarity task (positive vs. negative) using the part of our data labeled with S-POS and S-NEG tags. With the SVMs classifier, we find that the WH, FG, and MC models outperform the baseline (bow on DEV, acc $= 68.09\%$) on the DEV data with 71.63%, 73.05%, and 73.76% respectively, but not on TEST where performance of these models drops to the same accuracy of 65.67% with each of the three settings. We conclude that the structural and social context features we propose are better suited for learning subjective (but not polar) language, and so we do not proceed with further experiments with GRUs on the polarity task.

6. Recurrent Neural Networks Models

Our deep learning model is based on a gated neural network. We now further introduced this class of methods. For notation, we denote scalars with italic lowercase (e.g., *x*), vectors with bold lowercase (e.g.,**x**), and matrices with bold uppercase (e.g.,**W**).

Recurrent Neural Network A recurrent neural network (RNN) is a neural network architecture that, at each time step t, takes an input vector $\mathbf{x}_t \in \mathbb{R}^n$ and a hidden state vector $\mathbf{h}_{t-1} \in \mathbb{R}^m$ and produces the next hidden state $\mathbf{h}_t$ by applying the following recursive operation:

$$\mathbf{h}_t = f\left(\mathbf{W}\mathbf{x}_t + \mathbf{U}\mathbf{h}_{t-1} + \mathbf{b}\right) \tag{1}$$

Where the input to hidden matrix $\mathbf{W} \in \mathbb{R}^{m \times n}$, the hidden to hidden matrix $\mathbf{U} \in \mathbb{R}^{m \times m}$, and the bias vector $\mathbf{b} \in \mathbb{R}^m$ are parameters of an affine transformation and f is an element-wise nonlinearity. In theory, this design should enable an RNN to capture all historical information up to time step $\mathbf{h}_t$. In practice, however,RNNs suffer from the problems of vanishing/exploding gradients (Bengio et al., 1994; Pascanu et al., 2013) while trying to learn long-range dependencies.

LSTM Long short-term memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) are designed to addresse this very problem of learning long-term dependencies: LSTMs are basically a variation of RNNs that are augmented with a memory cell $\mathbf{c}_t \in \mathbb{R}^n$ at each time step. As such, in addition to the input vector $\mathbf{x}_t$, the hidden vector $\mathbf{h}_{t-1}$, an LSTM takes a cell state vector $\mathbf{c}_{t-1}$ and produces $\mathbf{h}_t$ and $\mathbf{c}_t$ via the calculations below:

$$\begin{aligned}
\mathbf{i}_t &= \sigma\left(\mathbf{W}^i x_t + \mathbf{U}^i \mathbf{h}_{t-1} + \mathbf{b}^i\right) \\
\mathbf{f}_t &= \sigma\left(\mathbf{W}^f x_t + \mathbf{U}^f \mathbf{h}_{t-1} + \mathbf{b}^f\right) \\
\mathbf{o}_t &= \sigma\left(\mathbf{W}^o x_t + \mathbf{U}^o \mathbf{h}_{t-1} + \mathbf{b}^o\right) \\
\mathbf{g}_t &= \tanh\left(\mathbf{W}^g x_t + \mathbf{U}^g \mathbf{h}_{t-1} + \mathbf{b}^g\right) \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)
\end{aligned} \tag{2}$$

Where $\sigma(\cdot)$ and $\tanh(\cdot)$ are the element-wise sigmoid and hyperbolic tangent functions, $\odot$ the element-wise multiplication operator, and $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{o}_t$ are the *input, forget*, and *output* gates. The $\mathbf{g}_t$ is a new memory cell vector with candidates that could be added to the state. The LSTM parameters $\mathbf{W}_j$, $\mathbf{U}_j$, and $\mathbf{b}_j$ are for $j \in \{i, f, o, g\}$.

GRUs (Cho et al., 2014; Chung et al., 2015) propose a variation of LSTM with a *reset gate* $\mathbf{r}_t$, an update state $\mathbf{z}_t$, and a new simpler hidden unit $\mathbf{h}_t$, as follows:

$$\begin{aligned}
\mathbf{r}_t &= \sigma\left(\mathbf{W}^r x_t + \mathbf{U}^r \mathbf{h}_{t-1} + \mathbf{b}^r\right) \\
\mathbf{z}_t &= \sigma\left(\mathbf{W}^z x_t + \mathbf{U}^z \mathbf{h}_{t-1} + \mathbf{b}^z\right) \\
\tilde{\mathbf{h}}_t &= \tanh\left(\mathbf{W} x_t + \mathbf{r}_t * \mathbf{U}^{\tilde{h}} \mathbf{h}_{t-1} + \mathbf{b}^{\tilde{h}}\right) \\
\mathbf{h}_t &= \mathbf{z}_t * \mathbf{h}_{t-1} + (1 - \mathbf{z}_t) * \tilde{\mathbf{h}}_t
\end{aligned} \tag{3}$$

The GRU parameters $\mathbf{W}_j$, $\mathbf{U}_j$, and $\mathbf{b}_j$ are for $j \in \{r, z, \tilde{h}\}$. In GRUs, the hidden state is forced to ignore a previous hidden state when the reset gate is close to 0, thus enabling the network to forget or drop irrelevant information. In addition, similar to an LSTM memory cell, the update gate controls how much information carries over from a previous hidden state to the current hidden state. GRUs are simpler and faster than LSTM, and so we use them instead of LSTMs in this work.

Network Architecture & Hyper-Parameters For GRUs, we use the same data split as described above with SVMs: 80% TRAIN, 10% DEV, and 10% TEST. We optimize the GRU hyper-parameters on the DEV set. We use a vocabulary size of $100K$ words, a word embedding vector of size 300 dimensions that we learn directly from the TRAIN, an input maximum length of 30 words, 2 epochs, and the Adam (Kingma and Ba, 2014) optimizer with a learning rate of 0.001. We use a GRU layer with 500 units input, followed by 3 dense layers each with $1,000$ units. To regularize the network, we use dropout (Hinton et al., 2012) with a dropout rate of 0.5 after the first dense layer. For our loss function, we use binary cross-entropy. We use a mini-batch (Cotter et al., 2011) size of 128.

6.1. Results

Table 4 shows the best results acquired with feature engineering using our SVM classifier on both DEV and TEST from the previous section. As Table 4 shows, our GRUs model achieve an accuracy of 77.66% on DEV. This is $\sim 15\%$ higher than our baseline (`base`). On TEST, the model achieves 77.19, which is 14.52% higher than the baseline. This gain on TEST is also 10.96% higher than an SVM bag-of-words (`bow`) classifier on the same TEST set. Compared to the best accuracy on TEST with SVMs (acquired both with WH and FG, both at 75.41%), not to our surprise GRUs are 1.78% higher. This, however, emphasizes the utility of our feature set with the SVMs approach. Interestingly, the SVM models are better when it comes to detecting the `SUBJ` class: On TEST, our best SVMs models are a whooping 41.51% F_1-score higher than GRUs. The same observation holds with the results on DEV as well, with $\sim 21\%$ edge for the SVM classifier. It can be immediately seen that improvements are possible by simply combining predictions from the models with both approaches in an ensemble set up. We cast further investigation in this direction as potentially promising future research.

Table 4: Results with Gated Recurrent Neural Networks

				OBJ			SUBJ		
	setting	**acc**	**avg-f**	**prec**	**rec**	**f**	**prec**	**rec**	**f**
	base (svm bow)	62.67	62.56	51.19	74.14	60.56	77.27	55.43	64.56
DEV	**MC**	70.00	69.28	59.42	70.69	64.57	79.01	69.57	73.99
	GRU	**77.66**	76.54	81.59	89.45	85.34	62.30	46.34	53.15
TEST	**bow (svm)**	66.23	65.69	54.67	70.09	61.42	77.42	63.83	69.97
	WH	75.41	73.96	68.10	67.52	67.81	79.89	80.32	80.11
	FG	75.41	73.96	68.10	67.52	67.81	79.89	80.32	80.11
	GRU	**77.19**	76.02	77.90	95.98	86.00	70.97	26.51	38.60

7. Conclusion

We described successful models for learning subjective language from the Twitter domain. For learning, we introduced a framework of structural and social context features and showed its utility in classification with an SVMs approach. More specifically, our rich feature set totals 30 individual features that we also organize thematically into 11 different groups. Further, we introduced two feature selection methods, a Monte Carlo (MC) method for picking the best combinations of individual features and another method for exhaustive feature group selection (FG). We also analyzed the performance of the different combinations of feature groups as well as the individual successful features on the task, with illustrative examples. Our best performing model with these hand-crafted features on the blind test set is $> 12\%$ higher than our baseline. In addition, we carefully developed a highly successful deep gated recurrent neural network classifier that yields $\sim 14.50\%$ accuracy gains over our baseline. Comparing the classical SVMs classifiers to the GRUs on the task, we show the utility of our rich feature set and identify a promising route for future research where these approaches can be combined. Other future directions include expanding our work to other domains and possibly other languages.

8. Acknowledgements

This work was partially funded by UBC Hampton Grant #12R74395 and UBC Work Learn Award to the author. The research was also enabled in part by support provided by WestGrid (`https://www.westgrid.ca/`) and Compute Canada (`www.computecanada.ca`).

9. Bibliographical References

Abdul-Mageed, M. and Herring, S. (2008). Arabic and english news coverage on aljazeera.net. In *Proceedings of Cultural Attitudes Towards Technology and Communication 2008 (CATaC'08), Nimes, France*.

Abdul-Mageed, M., Diab, M., and Kübler, S. (2014). Samar: Subjectivity and sentiment analysis for arabic social media. *Computer Speech & Language*, 28(1):20–37.

Abdul-Mageed, M., AlHuzli, H., and DuaaAbu Elhija, M. D. (2016). Dina: A multi-dialect dataset for arabic emotion analysis. In *The 2nd Workshop on Arabic Corpora and Processing Tools 2016 Theme: Social Media*, page 29.

Abdul-Mageed, M., Alhuzali, H., and Elaraby, M. (2018). You tweet what you speak: A city-level dataset of arabic dialects. In *LREC*.

Abdul-Mageed, M. (2017a). Modeling subjectivity and sentiment in lexical space. In *Submitted*.

Abdul-Mageed, M. (2017b). Not all segments are created equal: Syntactically motivated sentiment analysis in lexical space. *WANLP 2017 (co-located with EACL 2017)*, page 147.

Aly, M. A. and Atiya, A. F. (2013). Labr: A large scale arabic book reviews dataset. In *ACL (2)*, pages 494–498.

Androutsopoulos, J. and Beißwenger, M. (2008). Introduction: Data and methods in computer-mediated discourse analysis. *Language@ Internet*, 5(2):1–7.

Baccianella, S., Esuli, A., and Sebastiani, F. (2010). Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Seventh conference on International Language Resources and Evaluation, Malta. Retrieved May*, volume 25, page 2010.

Banfield, A. (1982). *Unspeakable Sentences: Narration and Representation in the Language of Fiction*. Routledge & Kegan Paul, Boston.

Benamara, F., Cesarano, C., Picariello, A., Reforgiato, D., and Subrahmanian, V. (2007). Sentiment analysis: Adjectives and adverbs are better than adjectives alone. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*.

Benamara, F., Chardon, B., Mathieu, Y. Y., Popescu, V., et al. (2011). Towards context-based subjectivity analysis. In *IJCNLP*, pages 1180–1188.

Benamara, F., Taboada, M., and Mathieu, Y. (2017). Evaluative language beyond bags of words: Linguistic insights and computational applications. *Computational Linguistics*.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.

Bieswanger, M. (2013). 19. micro-linguistic structural features of computer-mediated communication. *Pragmatics of computer-mediated communication*, 9:463.

Blunsom, P., Grefenstette, E., and Kalchbrenner, N. (2014). A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting*

of the Association for Computational Linguistics. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics.

Breck, E., Choi, Y., and Cardie, C. (2007). Identifying expressions of opinion in context. In *IJCAI*, volume 7, pages 2683–2688.

Brown, P. and Levinson, S. (1987). *Politeness: Some universals in language usage*, volume 4. Cambridge Univ Pr.

Burger, J. D., Henderson, J., Kim, G., and Zarrella, G. (2011). Discriminating gender on twitter. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1301–1309. Association for Computational Linguistics.

Chandrashekar, G. and Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Chung, J., Gülçehre, C., Cho, K., and Bengio, Y. (2015). Gated feedback recurrent neural networks. In *ICML*, pages 2067–2075.

Cohen, W. W. (1996). Learning trees and rules with set-valued features. In *AAAI/IAAI, Vol. 1*, pages 709–716.

Cotter, A., Shamir, O., Srebro, N., and Sridharan, K. (2011). Better mini-batch algorithms via accelerated gradient methods. In *Advances in neural information processing systems*, pages 1647–1655.

Cui, H., Mittal, V., and Datar, M. (2006). Comparative experiments on sentiment classification for online product reviews. In *AAAI*, volume 6, pages 1265–1270.

Dash, M. and Liu, H. (1997). Feature selection for classification. *Intelligent data analysis*, 1(3):131–156.

Deng, H., Han, J., Li, H., Ji, H., Wang, H., and Lu, Y. (2014). Exploring and inferring user–user pseudo-friendship for sentiment analysis with heterogeneous networks. *Statistical Analysis and Data Mining*, 7(4):308–321.

ElSahar, H. and El-Beltagy, S. R. (2015). Building large arabic multi-domain resources for sentiment analysis. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 23–34. Springer.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research*, 3(Mar):1289–1305.

Gamon, M. (2004). Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In *Proceedings of the 20th international conference on Computational Linguistics*, page 841. Association for Computational Linguistics.

Goldberg, Y. (2016). A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, 57:345–420.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT Press.

Hasan, K. S. and Ng, V. (2013). Extra-linguistic constraints on stance recognition in ideological debates. In *ACL (2)*, pages 816–821.

Hatzivassiloglou, V. and McKeown, K. R. (1997). Predicting the semantic orientation of adjectives. In *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*, pages 174–181. Association for Computational Linguistics.

Hatzivassiloglou, V. and Wiebe, J. M. (2000). Effects of adjective orientation and gradability on sentence subjectivity. In *Proceedings of the 18th conference on Computational linguistics-Volume 1*, pages 299–305. Association for Computational Linguistics.

Herring, S., Stein, D., and Virtanen, T. (2013). *Pragmatics of computer-mediated communication*, volume 9. Walter de Gruyter.

Herring, S. (1994). Gender differences in computer-mediated communication: Bringing familiar baggage to the new frontier. *Retrieved April*, 29:2002.

Herring, S. C. (2007). A faceted classification scheme for computer-mediated discourse. *Language@ internet*, 4(1):1–37.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Irsoy, O. and Cardie, C. (2014). Deep recursive neural networks for compositionality in language. In *Advances in Neural Information Processing Systems*, pages 2096–2104.

Joachims, T. (2002). Support vector machines. In *Learning to Classify Text Using Support Vector Machines*, pages 35–44. Springer.

Kennedy, A. and Inkpen, D. (2006). Sentiment classification of movie reviews using contextual valence shifters. *Computational intelligence*, 22(2):110–125.

Kessler, J. S. and Nicolov, N. (2009). Targeting sentiment expressions through supervised ranking of linguistic configurations. In *ICWSM*.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klenner, M., Fahrni, A., and Petrakis, S. (2009). Polart: A robust tool for sentiment analysis. In *Proceedings of the 17th Nordic Conference of Computational Linguistics*, volume 4, pages 235–238.

Labutov, I. and Lipson, H. (2013). Re-embedding words. In *ACL (2)*, pages 489–493.

Le, Q. V. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

Li, J., Luong, M.-T., Jurafsky, D., and Hovy, E. (2015).

When are tree structures necessary for deep learning of representations? *arXiv preprint arXiv:1503.00185*.

Lin, W.-H. and Hauptmann, A. (2006). Are these documents written from different perspectives?: a test of different perspectives based on statistical distribution divergence. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 1057–1064. Association for Computational Linguistics.

Liu, P., Qiu, X., Chen, X., Wu, S., and Huang, X. (2015). Multi-timescale long short-term memory neural network for modelling sentences and documents. In *EMNLP*, pages 2326–2335. Citeseer.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 142–150. Association for Computational Linguistics.

Mikolov, T., Yih, W.-t., and Zweig, G. (2013). Linguistic regularities in continuous space word representations. In *Hlt-naacl*, volume 13, pages 746–751.

Mitchell, J. and Lapata, M. (2010). Composition in distributional models of semantics. *Cognitive science*, 34(8):1388–1429.

Mohammad, S. M. and Kiritchenko, S. (2015). Using hashtags to capture fine emotion categories from tweets. *Computational Intelligence*, 31(2):301–326.

Montoyo, A., MartíNez-Barco, P., and Balahur, A. (2012). Subjectivity and sentiment analysis: An overview of the current state of the area and envisaged developments.

Mourad, A. and Darwish, K. (2013). Subjectivity and sentiment analysis of modern standard arabic and arabic microblogs. In *Proceedings of the 4th workshop on computational approaches to subjectivity, sentiment and social media analysis*, pages 55–64.

Ng, V., Dasgupta, S., and Arifin, S. (2006). Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 611–618. Association for Computational Linguistics.

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. *ICML (3)*, 28:1310–1318.

Persing, I. and Ng, V. (2014). Vote prediction on comments in social polls. In *EMNLP*, pages 1127–1138.

Purver, M. and Battersby, S. (2012). Experimenting with distant supervision for emotion classification. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 482–491. Association for Computational Linguistics.

Rao, D., Yarowsky, D., Shreevats, A., and Gupta, M. (2010). Classifying latent user attributes in twitter. In *Proceedings of the 2nd international workshop on Search and mining user-generated contents*, pages 37–44. ACM.

Ren, Y., Zhang, Y., Zhang, M., and Ji, D. (2016). Context-sensitive twitter sentiment classification using neural network. In *AAAI*, pages 215–221.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., Potts, C., et al. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642. Citeseer.

Tai, K. S., Socher, R., and Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.

Tan, C., Lee, L., Tang, J., Jiang, L., Zhou, M., and Li, P. (2011). User-level sentiment analysis incorporating social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1397–1405. ACM.

Tang, D., Wei, F., Qin, B., Zhou, M., and Liu, T. (2014a). Building large-scale twitter-specific sentiment lexicon: A representation learning approach. In *COLING*, pages 172–182.

Tang, D., Wei, F., Yang, N., Zhou, M., Liu, T., and Qin, B. (2014b). Learning sentiment-specific word embedding for twitter sentiment classification. In *ACL (1)*, pages 1555–1565.

Tang, D., Qin, B., and Liu, T. (2015). Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432.

Thomas, M., Pang, B., and Lee, L. (2006). Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 327–335. Association for Computational Linguistics.

Turney, P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics.

Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20:273–297.

Volkova, S., Wilson, T., and Yarowsky, D. (2013). Exploring demographic language variations to improve multilingual sentiment analysis in social media. In *EMNLP*, pages 1815–1827.

Volkova, S., Bachrach, Y., Armstrong, M., and Sharma, V. (2015). Inferring latent user properties from texts published in social media. In *AAAI*, pages 4296–4297.

Wang, W., Chen, L., Thirunarayan, K., and Sheth, A. P. (2012). Harnessing twitter "big data" for automatic emotion identification. In *Privacy, Security, Risk and Trust*

(PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom), pages 587–592. IEEE.

Wiebe, J., Wilson, T., Bruce, R., Bell, M., and Martin, M. (2004). Learning subjective language. *Computational linguistics*, 30(3):277–308.

Wiebe, J. (2000). Learning subjective adjectives from corpora. In *Proc.17th National Conference on Artificial Intelligence (AAAI-2000)*, pages 735–741, Austin, Texas.

Wiegand, M., Balahur, A., Roth, B., Klakow, D., and Montoyo, A. (2010). A survey on the role of negation in sentiment analysis. In *Proceedings of the workshop on negation and speculation in natural language processing*, pages 60–68. Association for Computational Linguistics.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics.

Wilson, T., Wiebe, J., and Hwa, R. (2006). Recognizing strong and weak opinion clauses. *Computational Intelligence*, 22(2):73–99.

Yang, Y. and Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *Icml*, volume 97, pages 412–420.

Zhang, X., Zhao, J., and LeCun, Y. (2015). Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Zhang, M., Zhang, Y., and Vo, D.-T. (2016). Gated neural networks for targeted sentiment analysis. In *AAAI*, pages 3087–3093.

Zhuang, L., Jing, F., and Zhu, X.-Y. (2006). Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 43–50. ACM.