The Contribution of Embeddings to Sentiment Analysis on YouTube

Moniek Nieuwenhuis
CLCG, University of Groningen
The Netherlands
m.l.nieuwenhuis@student.rug.nl

Malvina Nissim
CLCG, University of Groningen
The Netherlands
m.nissim@rug.nl

Abstract

We train a variety of embeddings on a large corpus of YouTube comments, and test them on three different tasks on both the English and the Italian portions of the SenTube corpus. We show that in-domain (YouTube) embeddings perform better than previously used generic embeddings, achieving state-of-the-art performance on most of the tasks. We also show that a simple method for creating sentiment-aware embeddings outperforms previous strategies, and that sentiment embeddings are more informative than plain embeddings for the SenTube tasks.

1 Introduction and Background

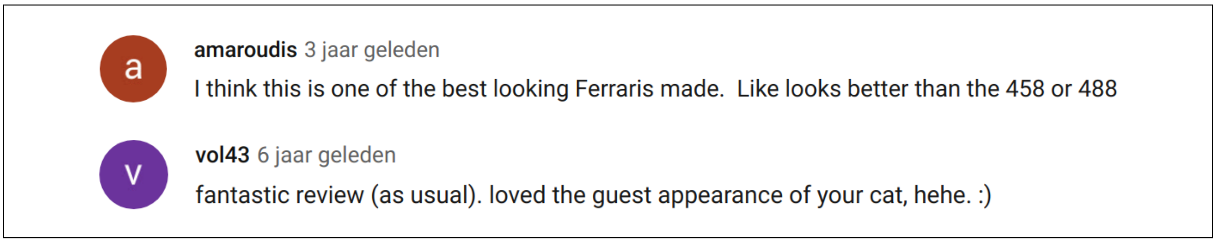
Sentiment analysis, or opinion mining, on social media is by now a well established task, though surely not solved (Liu et al., 2005; Barnes et al., 2017). Part of the difficulty comes from its intrinsic subjective nature, which makes creating reliable resources hard (Kiritchenko and Mohammad, 2017). Another part comes from its heavy interaction with pragmatic phenomena such as irony and world knowledge (Nissim and Patti, 2017; Basile et al., 2018; Cignarella et al., 2018; Van Hee et al., 2018). And another difficulty comes from the fact that given a piece of text, be it a tweet, or a review, it isn't always clear what exactly the expressed sentiment (should there be any) is about. In commercial reviews, for example, the target of a user's evaluation could be a specific aspect or part of a given product. Aspect-based sentiment analysis has developed as a subfield to address this problem (Thet et al., 2010; Pontiki et al., 2014).

The SenTube corpus (Uryupina et al., 2014) has been created along these lines. It contains English and Italian commercial or review videos about some product, and annotated comments. The annotations specify both the polarity (positive, negative, neutral) and the target (the video itself or the product in the video). In Figure 1 we show two positive comments with different targets.

The SenTube's tasks have been firstly addressed by Severyn et al. (2016) with an SVM based on topic and shallow syntactic information, later outperformed by a convolutional N-gram BiLSTM word embedding model (Nguyen and Le Nguyen, 2018). The corpus has also served as testbed for multiple state-of-the-art sentiment analysis methods (Barnes et al., 2017), with best results obtained using sentiment-specific word embeddings (Tang et al., 2014). On the English sentiment task of SenTube though this method does not outperform corpus-specific approaches (Severyn et al., 2016; Nguyen and Le Nguyen, 2018).

We further explore the potential of (sentiment) embeddings, using the model developed by Nguyen and Le Nguyen (2018). We believe that training in-domain (YouTube) embeddings rather than using generic ones might yield improvements, and that additional gains might come from sentiment-aware embeddings. In this context, we propose a simple new semi-supervised method to train sentiment embeddings and show that it performs better than two other existing ones. We run all experiments on English and Italian data.

Contributions We show that in-domain embeddings outperform generic embeddings on most task of the SenTube corpus for both Italian and English. We also show that sentiment embeddings obtained through a simple semi-supervised strategy that we newly introduce in this paper add a boost to performance. We make all developed Italian and English embeddings avail-



amaroudis 3 jaar geleden

I think this is one of the best looking Ferraris made. Like looks better than the 458 or 488

vol43 6 jaar geleden

fantastic review (as usual). loved the guest appearance of your cat, hehe. :)

Figure 1: Two sample comments on a video about a Ferrari car. Top: positive comment about the product. Bottom: positive comment about the video.

able at this link: `https://github.com/malvinanissim/youtube-embeds`.

2 Data and Task

We use two different datasets of YouTube comments. The first is the existing SenTube corpus (Uryupina et al., 2014). The other dataset is collected from YouTube to create a big semi-supervised corpus for making the embeddings.

2.1 SenTube corpus

The SenTube corpus contains 217 videos in English and 198 in Italian (Uryupina et al., 2014). All videos are a review or commercial about a product in the category "automobile" or "tablet".

All comments from the videos are annotated according to their target (whether they are about the video or about the product) and their sentiment polarity (positive, negative, neutral). Some of the comments were discarded because of spam, because they were written in a language other than the intended one (Italian for the Italian corpus, English for the English one), or just off topic. Sentiment is type-specific, and the following labels are used: *positive-product*, *negative-product*, *positive-video* and *negative-video*. If neither positive or negative is annotated, the comment is assumed to be *neutral*.

The corpus lends itself to three different tasks, all of which we tackle in this work:

- the *sentiment task*, namely predicting whether a YouTube comment is written in a positive, negative or a neutral sentiment.
- the *type task*, namely predicting if the comment is written about the product mentioned in the video, about the video itself or if it is not an informative comment (spam or off-topic).
- the *full task*: predicting at the same time the sentiment and the type of each comment.

From SenTube we exclude any comment that is annotated both as product-related and video-related or is both positive and negative. Table 1 shows the label distribution for the three tasks. All comments are further lowercased and tokenised.

2.2 Semi-supervised YouTube corpus

To train in-domain embeddings we collected more data from YouTube. We searched for relevant videos querying the YouTube API with a set of keywords ("car", "tablet", "macchina", "automobile", ...). For each retrieved video we checked that it was not already included in the SenTube corpus, and verified that its description was in English/Italian using Python's `langdetect` module. We then retrieved all comments for each video that had more than one comment.

Next, we used the convolutional N-gram BiLSTM word embedding model by (Nguyen and Le Nguyen, 2018), which has state-of-the-art performance on SenTube, to label the data on the sentiment task, as we want to exploit the labels to train sentiment embeddings. Table 2 shows an overview of the collected dataset. A manual check on a randomly chosen test set of 100 comments for each language, revealed a rough accuracy of just under 60% for English, and just under 65% for Italian.

3 Embeddings

We test three different categories of embeddings: some pre-trained models, a variety of models trained on our in-domain dataset, and sentiment-aware embeddings, which we obtain in three different ways. All of the embeddings are tested in the model developed by (Nguyen and Le Nguyen, 2018) to specifically tackle the SenTube tasks.

3.1 Plain Embeddings

Generic models For English we used GoogleNews vectors[1], which are those used in (Nguyen

[1] `https://code.google.com/archive/p/word2vec/`

Table 1: Label distribution for each task in the SenTube corpus

	English				Italian			
	Automobile	**%**	**Tablet**	**%**	**Automobile**	**%**	**Tablet**	**%**
Product-related	5,834	38.8	11,067	56.2	1,718	40.9	2,976	61.0
Video-related	5,201	34.5	3,665	18.6	1,317	31.4	845	17.3
Uninfo.	4,020	26.7	4,961	25.2	1,161	27.7	1,055	21.6
Positive sentiment	3,284	21.8	3,637	18.5	946	22.5	770	15.8
Negative sentiment	1,988	13.2	3,038	15.4	752	17.9	825	16.9
No sentiment/neutral	9,801	65.0	13,021	66.1	2,499	59.5	3,281	67.3
Product-pos.	1,740	11.5	2,280	11.6	479	11.4	544	11.4
Product-neg.	1,360	9.0	2,473	12.5	538	12.8	711	14.6
Product-neu.	2,744	18.2	6,310	32.0	703	16.8	1,721	35.3
Video-pos.	1,543	10.2	1,357	6.9	467	11.1	226	4.6
Video-neg.	628	4.2	565	2.9	214	5.1	114	2.3
Video-neu.	3,030	20.1	1,743	8.8	635	15.1	505	10.4
Uninfo.	4,028	26.7	4,968	25.2	1,161	27.7	1055	21.6

Table 2: Overview of extra data collected from YouTube

	English			Italian		
	Automobile	**Tablet**	**Total**	**Automobile**	**Tablet**	**Total**
Videos	1,592	1,675	3,267	1,622	1,151	2,773
Comments	1,028,136	587,506	1,615,642	99,328	118,274	217.602
Tokens	18,124,184	9,156,324	27,280,508	1,596,190	1,579,591	3,175,781
Unique tokens	754,962	416,835	1,030,574	170,956	155,738	277,114
Positive sentiment	165,725	97,439	263,164 (16.3%)	11,091	13,356	24,447(11.2%)
Negative sentiment	49,490	53,557	103,047 (6.4%)	4,898	4,514	9,412(4.3%)
Neutral sentiment	812,921	436,510	1,249,431 (77.3%)	83,339	100,404	183,743(84.4%)

and Le Nguyen, 2018), and the 200-dimensional GloVe Twitter embeddings[2]. For Italian we used vectors from (Bojanowski et al., 2016) a FastText model trained on the the Italian Wikipedia, and also used by (Nguyen and Le Nguyen, 2018). Furthermore, we tested two models developed at ISTI-CNR, which are trained on Italian Wikipedia with skip-gram's Word2Vec and with GloVe.[3]

In-domain trained models We trained three Word2Vec models (Mikolov et al., 2013), all of dimension 300, using Gensim (Řehůřek and Sojka, 2010). Beside a CBOW model with default settings, we trained two different skip-gram models, one with default settings and one with a negative sampling of 10. We also trained a FastText model (Bojanowski et al., 2016), and a 100-dimension GloVe model (Pennington et al., 2014).

3.2 Sentiment-aware Embeddings

We use three methods for adding sentiment to the embeddings, in all cases using the Word2Vec skip-gram models (Mikolov et al., 2013) with and without negative sampling 10. The first two methods are existing methods, namely retrofitting (Faruqui et al., 2015) and the refinement method suggested by Yu et al. (2017), while the third method is newly proposed in this work.

Retrofitting Retrofitting embedding models is a method to refine vector space representations using relational information from semantic lexicons by encouraging linked words to have similar vector representations (Faruqui et al., 2015).[4] We used two sentiment lexicons to retrofit the skip-gram models. A SentiWordNet-derived lexicon for English (Baccianella et al., 2010), and Sentix for Italian (Basile and Nissim, 2013).[5]

[2] https://nlp.stanford.edu/projects/glove/

[3] http://hlt.isti.cnr.it/wordembeddings/

[4] https://github.com/mfaruqui/retrofitting.

[5] http://valeriobasile.github.io/twita/sentix.html

Sentiment Embedding refinement We tested the method proposed by Yu et al. (2017) using the provided code[6] to refine our own skip-gram Word2Vec models. In this method the similar top-k words will be re-ranked by sentiment on the difference in valence scores from a sentiment lexicon. For English we used the E-ANEW sentiment lexicon (Warriner et al., 2013) and for Italian we used Sentix (Basile and Nissim, 2013).

Our Embedding refinement For each language, we use a sentiment lexicon and our YouTube corpus to train sentiment embeddings.

From the sentiment lexicon we create two lists of words: positive words (positive score > 0.6 and negative score < 0.2) and negative words (negative score > 0.6 and positive score < 0.2).

For each word in the positive list, we check if it occurs in a comment with a positive label. We do the same for the negative list and negative labelled comments. If the word occurs in the list we add the affixes `"_pos"` or `"_neg"` to the word occurrence in a positive or negative comment. If a word from the positive list is found in a comment with negative or neutral label it isn't touched, and likewise for words in the negative list. An example of this approach is in Table 3.

Example	Label
*"I **love_pos** this review! It's not the technical review that every YouTube vid has bit more of a usable hands on one! makes me **really_pos** want one even more than before! Thank you!"*	positive
*"I **love** being a cheapskate. Please tell me what in the world "gimp" is."*	neutral
*"I don't understand why people **love** apple shit [...]*	negative

Table 3: Example of the word "love" changed in the positive comment and not changed in neutral or negative comments.

We then trained the embeddings with skip-gram Word2Vec (Mikolov et al., 2013), with therein the two separate appearances of words, i.e. with and without affixes. This of course poses a problem at test time, since two vectors are now available for some of the words (`great_pos` and `great` for "great", for example, or `brutto_neg` and `brutto` for "brutto" [*en: ugly*]), but one must eventually choose one for representing the encountered word "great", or "brutto".

Instead of devising a strategy for choosing one of the two vectors, we opted for *re-joining* the two versions of the word into a single one, testing two different methods:

- *averaging*: average the vectors with each other; the two contexts have equal weight;
- *weighting*: weigh each vector by the proportion of times the word is in either context (in the semi-supervised corpus), and sum them.

4 Experiments

We split the SenTube corpus in 50% train and 50% test. We could not exactly replicate the split by Nguyen and Le Nguyen (2018) due to lack of sufficient details in their code. We use their model to test all embeddings, including those used in their implementation (GoogleNews for English, and FastText for Italian), for direct comparison with our embeddings. For completeness, we also include the results reported by Severyn et al. (2016) (with their own split), and a most frequent label baseline for each task. As was done in previous work on this corpus, and for more direct comparison, we report accuracy across all experiments.

Table 4: English embeddings results

Task	Embeddings	AUTO	TABLET
Sentiment	Most frequent label baseline	0.632	0.680
	(Severyn et al., 2016)	0.557	0.705
	(Nguyen and Le Nguyen, 2018)	0.669	0.702
in-domain	CBOW	0.725	0.755
	SKIP	**0.740**	0.750
	SKIP neg samp	0.730	**0.756**
	GloVe	0.709	0.754
	FastText	0.729	0.754
generic	GoogleNews	0.715	0.748
	GLoVe Twitter	0.723	0.742
Type	Most frequent label baseline	0.384	0.565
	(Severyn et al., 2016)	0.594	0.786
	(Nguyen and Le Nguyen, 2018)	0.684	0.795
in-domain	CBOW	0.714	0.784
	SKIP	**0.733**	0.800
	SKIP neg samp	0.723	**0.801**
	GloVe	0.697	0.779
	FastText	0.727	0.779
generic	GoogleNews	0.688	0.773
	GLoVe Twitter	0.690	0.775
Full	Most frequent label baseline	0.243	0.342
	(Severyn et al., 2016)	0.415	0.603
	(Nguyen and Le Nguyen, 2018)	0.538	0.613
in-domain	CBOW	0.536	0.618
	SKIP	0.547	0.621
	SKIP neg samp	**0.558**	**0.629**
	GloVe	0.504	0.596
	FastText	0.540	0.615
generic	GoogleNews	0.504	0.580
	GLoVe Twitter	0.487	0.600

[6] `https://github.com/wangjin0818/word_embedding_refine`

4.1 Results with plain embeddings

The results using plain embeddings are shown in Tables 4 and 5. Most of the in-domain embeddings on English outperform the GoogleNews vectors used by Nguyen and Le Nguyen (2018); the results are also higher than those reported in previous work with different splits (Severyn et al., 2016; Nguyen and Le Nguyen, 2018). Only for both full tasks and the tablet type task there are a few of the in-domain embeddings which do not outperform on previous work results. For Italian, not all in-domain embeddings outperform previous work in all tasks, but they mostly do when embeddings used in previous work are tested on the same split. For both languages the skip-gram models are performing best compared to all the other in-domain embedding models. On Italian, the generic Wikipedia SKIP embeddings and the generic FastText embeddings (Bojanowski et al., 2016) are performing slightly better on the sentiment and full task for tablets.

Table 5: Italian embedding results

Task	Embeddings	AUTO	TABLET
Sentiment	Most frequent label baseline	0.601	0.668
	(Severyn et al., 2016)	0.616	0.644
	(Nguyen and Le Nguyen, 2018)	0.614	0.656
	CBOW	0.622	0.700
	SKIP	0.636	0.687
in-domain	SKIP neg samp	**0.652**	0.697
	GloVe	0.607	0.673
	FastText	0.640	0.645
	FastText	0.648	0.682
generic	Wikipedia SKIP	0.629	**0.701**
	Wikipedia GloVe	0.613	0.679
Type	Most frequent label baseline	0.415	0.568
	(Severyn et al., 2016)	0.707	0.773
	(Nguyen and Le Nguyen, 2018)	0.748	**0.796**
	CBOW	0.742	0.710
	SKIP	**0.768**	0.695
in-domain	SKIP neg samp	0.762	**0.722**
	GloVe	0.744	0.676
	FastText	0.703	0.703
	FastText	0.769	0.716
generic	Wikipedia SKIP	0.756	0.682
	Wikipedia GloVe	0.725	0.694
Full	Most frequent label baseline	0.320	0.252
	(Severyn et al., 2016)	0.456	0.524
	(Nguyen and Le Nguyen, 2018)	0.511	**0.550**
	CBOW	0.470	0.484
	SKIP	0.489	0.487
in-domain	SKIP neg samp	**0.517**	0.485
	GloVe	0.450	0.490
	FastText	0.459	0.484
	FastText	0.491	**0.497**
generic	Wikipedia SKIP	0.492	0.495
	Wikipedia GloVe	0.441	0.449

4.2 Results with sentiment embeddings

Tables 6 and 7 show the results of the sentiment embeddings. In almost all tasks the sentiment embeddings outperform the plain embeddings. Surprisingly, this is true even for the English type task, while the sentiment automobile task has a slightly lower accuracy. For Italian only in the automobile type task sentiment embeddings do not outperform standard ones. Among the sentiment embeddings, our refinement method seems to work best, while retrofitting does not lead to any improvement.

In terms of weighing versus averaging the vectors in our method, for English averaging yields the best score three times, and weighting two times. For Italian, weighting yields the best result two times on the tablet data set, while for the full task averaging is better. For cars, weighting is better, but does not outperform plain embeddings.

Table 6: English sentiment embedding test

Task	Embeddings	AUTO	TABLET
Sentiment	SKIP neg samp retrofitted	0.701	0.751
	SKIP retrofitted	0.710	0.742
	SKIP sentiment embedding refinement	0.725	0.747
	SKIP neg samp sentiment embedding refinement	0.725	0.753
	SKIP sentiment change average	0.715	0.760
	SKIP sentiment change weight sum	0.737	**0.767**
	SKIP neg samp sentiment change average	0.729	0.758
	SKIP neg samp sentiment change weight sum	0.734	0.749
Type	SKIP neg samp retrofitted	0.688	0.774
	SKIP retrofitted	0.680	0.781
	SKIP sentiment embedding refinement	0.732	0.794
	SKIP neg samp sentiment embedding refinement	0.735	0.796
	SKIP sentiment change average	0.723	0.806
	SKIP sentiment change weight sum	0.716	0.798
	SKIP neg samp sentiment change average	0.722	**0.807**
	SKIP neg samp sentiment change weight sum	**0.739**	0.794
Full	SKIP neg samp retrofitted	0.500	0.600
	SKIP retrofitted	0.501	0.594
	SKIP sentiment embedding refinement	0.537	0.594
	SKIP neg samp sentiment embedding refinement	0.522	0.606
	SKIP sentiment change average	**0.560**	0.616
	SKIP sentiment change weight sum	0.544	0.623
	SKIP neg samp sentiment change average	0.549	**0.631**
	SKIP neg samp sentiment change weight sum	0.547	0.618

5 Conclusion

We have explored the contribution of in-domain embeddings on the SenTube corpus, on two domains and two languages. In 10 out of the 12 tasks, in-domain embeddings outperform generic ones. This confirms the experiments on the SENTIPOLC 2016 tasks (Barbieri et al., 2016) reported by Petrolito and Dell'Orletta (2018), who recommend the use of in-domain embeddings for sentiment analysis, especially if trained at the word rather than carachter level. However, a similar work in the field of sentiment analysis for soft-

Table 7: Italian sentiment embedding test

Task	Embeddings	AUTO	TABLET
Sentiment	SKIP neg samp retrofitted	0.649	0.682
	SKIP retrofitted	0.622	0.686
	SKIP sentiment embedding refinement	0.610	0.682
	SKIP neg samp sentiment embedding refinement	0.632	0.703
	SKIP sentiment change average	0.628	0.690
	SKIP sentiment change weight sum	0.623	0.704
	SKIP neg samp sentiment change average	0.640	0.682
	SKIP neg samp sentiment change weight sum	0.631	**0.710**
Type	SKIP neg samp retrofitted	0.730	0.712
	SKIP retrofitted	0.744	0.712
	SKIP sentiment embedding refinement	0.761	0.716
	SKIP neg samp sentiment embedding refinement	0.754	0.712
	SKIP sentiment change average	0.763	0.701
	SKIP sentiment change weight sum	0.746	0.729
	SKIP neg samp sentiment change average	0.760	0.732
	SKIP neg samp sentiment change weight sum	0.756	**0.739**
Full	SKIP neg samp retrofitted	0.478	0.447
	SKIP retrofitted	0.490	0.469
	SKIP sentiment embedding refinement	0.504	0.497
	SKIP neg samp sentiment embedding refinement	0.466	0.500
	SKIP sentiment change average	0.503	**0.512**
	SKIP sentiment change weight sum	0.505	0.477
	SKIP neg samp sentiment change average	0.497	0.489
	SKIP neg samp sentiment change weight sum	0.485	0.497

ware engineering texts, where in-domain (Stackoverflow) embeddings were compared to generic ones (GoogleNews), did not yield such clearcut results (Biswas et al., 2019).

We have also suggested a simple strategy to train sentiment embeddings, and shown that it outperforms other existing methods for this task. More in general, sentiment embeddings perform consistently better than plain embeddings for both languages in the ”tablet” domain, but less evidently so in the automobile domain. The reason for this requires further investigation. Further testing is also necessary to assess the influence of vector size in our experiments. Indeed, not all embeddings are trained with the same dimensions, an aspect that might also affect performance differences, though the true impact of size is not yet fully understood (Yin and Shen, 2018).

In terms of different embeddings types, it would be also interesting to compare our simple embedding refinement method, which takes specific contextual occurrences into account, with the performance of contextual word embeddings (Peters et al., 2018; Devlin et al., 2019), which work directly at the token rather than the type level. More complex training strategies could also be explored (Dong and De Melo, 2018).

Acknowledgments

We would like to thank the Center for Information Technology of the University of Groningen for providing access to the Peregrine high performance computing cluster which we used to run the experiments reported in this paper. We are also grateful to the reviewers for helpful comments.

References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. volume 10, 01.

Francesco Barbieri, Valerio Basile, Danilo Croce, Malvina Nissim, Nicole Novielli, and Viviana Patti. 2016. Overview of the EVALITA 2016 sentiment polarity classification task (SENTIPOLC). In *Proceedings of the 5th evaluation campaign of natural language processing and speech tools for Italian (EVALITA 2016)*.

Jeremy Barnes, Roman Klinger, and Sabine Schulte im Walde. 2017. Assessing state-of-the-art sentiment models on state-of-the-art sentiment datasets. *arXiv preprint arXiv:1709.04219*.

Valerio Basile and Malvina Nissim. 2013. Sentiment analysis on italian tweets. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 100–107.

Valerio Basile, Nicole Novielli, Danilo Croce, Francesco Barbieri, Malvina Nissim, and Viviana Patti. 2018. Sentiment polarity classification at evalita: Lessons learned and open challenges. *IEEE Transactions on Affective Computing*.

Eeshita Biswas, K Vijay-Shanker, and Lori Pollock. 2019. Exploring word embedding techniques to improve sentiment analysis of software engineering texts. In *Proceedings of the 16th International Conference on Mining Software Repositories*, pages 68–78. IEEE Press.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *CoRR*, abs/1607.04606.

Alessandra Teresa Cignarella, Simona Frenda, Valerio Basile, Cristina Bosco, Viviana Patti, Paolo Rosso, et al. 2018. Overview of the evalita 2018 task on irony detection in italian tweets (ironita). In *Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian (EVALITA 2018)*, volume 2263, pages 1–6. CEUR-WS.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Xin Dong and Gerard De Melo. 2018. A helping hand: Transfer learning for deep sentiment analysis. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2524–2534.

Manaal Faruqui, Jesse Dodge, Sujay K. Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of NAACL*.

Svetlana Kiritchenko and Saif M Mohammad. 2017. Capturing reliable fine-grained sentiment associations by crowdsourcing and best-worst scaling. *arXiv preprint arXiv:1712.01741*.

Bing Liu, Minqing Hu, and Junsheng Cheng. 2005. Opinion observer: analyzing and comparing opinions on the web. In *Proceedings of the 14th international conference on World Wide Web*, pages 342–351. ACM.

Tomas Mikolov, Kai Chen, G.s Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Proceedings of Workshop at ICLR*, 2013, 01.

Huy Tien Nguyen and Minh Le Nguyen. 2018. Multilingual opinion mining on youtube–a convolutional n-gram bilstm word embedding. *Information Processing & Management*, 54(3):451–462.

Malvina Nissim and Viviana Patti. 2017. Semantic aspects in sentiment analysis. In *Sentiment analysis in social networks*, pages 31–48. Elsevier.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 2227–2237.

Ruggero Petrolito and Felice Dell'Orletta. 2018. Word embeddings in sentiment analysis. In *CLiC-it*.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May. ELRA. `http://is.muni.cz/publication/884893/en`.

Aliaksei Severyn, Alessandro Moschitti, Olga Uryupina, Barbara Plank, and Katja Filippova. 2016. Multi-lingual opinion mining on youtube. *Information Processing & Management*, 52(1):46–60.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1555–1565. Association for Computational Linguistics.

Tun Thura Thet, Jin-Cheon Na, and Christopher SG Khoo. 2010. Aspect-based sentiment analysis of movie reviews on discussion boards. *Journal of information science*, 36(6):823–848.

Olga Uryupina, Barbara Plank, Aliaksei Severyn, Agata Rotondi, and Alessandro Moschitti. 2014. Sentube: A corpus for sentiment analysis on youtube social media. In *LREC*, pages 4244–4249.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. Semeval-2018 task 3: Irony detection in english tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 39–50.

Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. 2013. Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior Research Methods*, 45(4):1191–1207, Dec.

Zi Yin and Yuanyuan Shen. 2018. On the dimensionality of word embedding. In *Advances in Neural Information Processing Systems*, pages 887–898.

Liang-Chih Yu, Jin Wang, K Lai, and Xuejie Zhang. 2017. Refining word embeddings for sentiment analysis. pages 534–539, 01.