Proceedings of EMSASW2018 - 4th Workshop on Sentic Computing, Sentiment Analysis, Opinion Mining, and Emotion Detection

Co-located with ESWC 2018
15th European Semantic Web Conference
Heraklion, Crete, Greece
4th June 2018

Edited by

Diego Reforgiato Recupero *
Mauro Dragoni **
Davide Buscaldi ***
Mehwish Alam ****
Erik Cambria *****

∗ University of Cagliari, Cagliari, Italy
∗∗ Fondazione Bruno Kessler, Trento, Italy
∗ ∗ ∗ Universitè Paris 13, USPC, Paris, France
∗ ∗ ∗∗ STLab ISTC-CNR, Rome, Italy
∗ ∗ ∗ ∗ ∗ Nanyang Technological University, Singapore

Organizing Committee

- Diego Reforgiato Recupero, University of Cagliari, Cagliari, Italy
- Mauro Dragoni, Fondazione Bruno Kessler, Trento, Italy
- Davide Buscaldi, Universitè Paris 13, USPC, Paris, France
- Mehwish Alam, STLab ISTC-CNR, Rome, Italy
- Erik Cambria, Nanyang Technological University, Singapore

Program Committee

- Rada Mihalcea, University of North Texas (USA)
- Ping Chen, University of Houston-Downtown (USA)
- Yongzheng Zhang, LinkedIn Inc. (USA)
- Giuseppe Di Fabbrizio, Amazon Inc. (USA)
- Soujanya Poria, Nanyang Technological University (Singapore)
- Yunqing Xia, Tsinghua University (China)
- Rui Xia, Nanjing University of Science and Technology (China)
- Jane Hsu, National Taiwan University (Taiwan)
- Rafal Rzepka, Hokkaido University (Japan)
- Amir Hussain, University of Stirling (UK)
- Alexander Gelbukh, National Polytechnic Institute (Mexico)
- Bjoern Schuller, Technical University of Munich (Germany)
- Amitava Das, Samsung Research India (India)
- Dipankar Das, National Institute of Technology (India)
- Stefano Squartini, Marche Polytechnic University (Italy)
- Cristina Bosco, University of Torino (Italy)
- Paolo Rosso, Technical University of Valencia (Spain)

Preface

As the Web rapidly evolves, people are becoming increasingly enthusiastic about interacting, sharing, and collaborating through social networks, online communities, blogs, wikis, and the like. In recent years, this collective intelligence has spread to many different areas, with particular focus on fields related to everyday life such as commerce, tourism, education, and health, causing the size of the social Web to expand exponentially.

To identify the emotions (e.g. sentiment polarity, sadness, happiness, anger, irony, sarcasm, etc.) and the modality (e.g. doubt, certainty, obligation, liability, desire, etc.) expressed in this continuously growing content is critical to enable the correct interpretation of the opinions expressed or reported about social events, political movements, company strategies, marketing campaigns, product preferences, etc.

This has raised growing interest both within the scientific community, by providing it with new research challenges, as well as in the business world, as applications such as marketing and financial prediction would gain remarkable benefits.

One of the main application tasks in this context is opinion mining [1], which is addressed by a significant number of Natural Language Processing techniques, e.g. for distinguishing objective from subjective statements [2], as well as for more fine-grained analysis of sentiment, such as polarity and emotions [9]. Recently, this has been extended to the detection of irony, humor, and other forms of figurative language [3]. In practice, this has led to the organisation of a series of shared tasks on sentiment analysis, including irony and figurative language detection (SemEval 2013, 2014, 2015, 2018), sometimes focused on the domain of financial technology [25, 26, 27, 28] with the production of annotated data and development of running systems. A similar challenge for irony polarity detection has been proposed for the Italian language at SENTIPOLC[1], indicating a growing interest about irony detection in the international NLP community. Similar challenges, not involving directly an irony detection task, but in which irony detection may prove useful, have been organized also for French (DEFT2015[2]) and Spanish (TASS2015[3]). In [10], the authors propose an algorithm for irony detection based on semantic similarity. Other studies such as [11, 12, 13, 14] consider features such as ambiguity, polarity etc.. However, the later also relies on decision trees.

However, existing solutions still have many limitations leaving the challenge of emotions and modality analysis still open. For example, there is the need for building/enriching semantic/cognitive resources for supporting emotion and modality recognition and analysis. Additionally, the joint treatment of modality and emotion is, computationally, trailing behind, and therefore the focus of ongoing, current research. Also, while we can produce rather robust deep semantic analysis of natural language, we still need to tune this analysis to-

[1] http://www.di.unito.it/~tutreeb/sentipolc-evalita14/

[2] https://deft.limsi.fr/2015/

[3] https://gplsi.dlsi.ua.es/sepln15/en/node/36

wards the processing of sentiment and modalities, which cannot be addressed by means of statistical models only, currently the prevailing approaches to sentiment analysis in NLP. The hybridization of NLP techniques with Semantic Web technologies is therefore a direction worth exploring, as recently shown in [4, 6, 7, 8, 5, 17, 21, 24, 23, 22].

This workshop intends to be a discussion forum gathering researchers and industries from Cognitive Linguistics, NLP, Machine Learning, Semantic Web, Big Data, and related areas for presenting their ideas on the relation between Semantic Web and the study of emotions and modalities.

Opinion mining, sentiment analysis, analysis of emotions and modalities are popular topics in the Natural Language Processing and Linguistics research fields. Regular workshops and challenges (shared tasks) on these themes are organised as co-located events with major conferences, such as IJCAI and ACL. Another recently organised related event is the MOMA (Models for Modality Annotation), a workshop held in London (April 2015) in conjunction with the International Conference on Computational Semantics (IWCS 2015). Our workshop intends to complement these events, focusing on the relation between these topics and the Semantic Web.

References

[1] Bo, P., and Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval , 2 (1-2), 1-135.

[2] Wiebe, J., and Ellen, R. (2005). Creating Subjective and Objective Sentence Classifiers from Unannotated Texts. Computational Linguistics and Intelligent Text Processing 6th International Conference, CICLing (pp. 486-497). Mexico City: Springer.

[3] Paula, C., Sarmento, L., Silva, M. J., and de Oliveira, E. (2009). Clues for detecting irony in user-generated contents: oh...!! it's so easy;-). Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion (pp. 53-56). ACM.

[4] Diego Reforgiato Recupero, Valentina Presutti, Sergio Consoli, Aldo Gangemi, Andrea Giovanni Nuzzolese: Sentilo: Frame-Based Sentiment Analysis. Cognitive Computation 7(2): 211-225 (2015)

[5] Aldo Gangemi, Valentina Presutti, Diego Reforgiato Recupero: Frame-Based Detection of Opinion Holders and Topics: A Model and a Tool. IEEE Comp. Int. Mag. 9(1): 20-30 (2014)

[6] Saif, H., He, Y., and Alani, H. (2012). Semantic sentiment analysis of Twitter. 11th International Semantic Web Conference (ISWC 2012) (pp. 508-524). Springer.

[7] Gangemi, A., Presutti, V., and Reforgiato Recupero, D. (2014). Frame-based detection of opinion holders and topics: a model and a tool. IEEE Computational Intelligence , 9 (1), 20-30.

[8] Cambria, E., and Hussain, A. (2012). Sentic Computing: Techniques, Tools, and Applications. Springer.

[9] Liu, B. (2012). Sentiment Analysis and Opinion Mining. Synthesis Lectures on Human Language Technologies. Chicago: Morgan & Claypool Publishers.

[10] Toropova, V., A., (2014). Irony detection based on semantic similarity. SPIIRAS Proceedings, Vol. 1.

[11] Reyes, A. and Rosso, P. and Buscaldi, D (2012). From humor recognition to irony detection: The figurative language of social media. Journal of Data & Knowledge Engineering, Vol. 74.

[12] Reyes, A. and Rosso, P. and Veale, T., (2013). A multidimensional approach for detecting irony in Twitter. Language Resources and Evaluation

[13] Barbieri, F. and Saggion, H., (2014). Automatic Detection of Irony and Humour in Twitter. International Conference on Computational Creativity.

[14] Recupero D. R., Alam, M. Buscaldi, D., Grezka, A., Tavazoee, F., 2017. Figurative language detection in Social Media leveraging Semantic Frames and BabelNet Synsets. Journal of Computational Linguistics (Under Review)

[15] Dragoni, M. (2017). A Three-Phase Approach for Exploiting Opinion Mining in Computational Advertising. IEEE Intelligent Systems 32(3): 21-27 (2017)

[16] Dragoni, M., Tettamanzi, A.G.B., da Costa Pereira, C. (2015). Propagating and Aggregating Fuzzy Polarities for Concept-Level Sentiment Analysis. Cognitive Computation 7(2): 186-197 (2015)

[17] Dragoni, M., Petrucci, G. (2017). A Neural Word Embeddings Approach For Multi-Domain Sentiment Analysis. IEEE Transactions on Affective Computing 8(4): 457-470 (2017)

[18] Mauro Dragoni, Diego Reforgiato Recupero: Proceedings of the 3rd International Workshop at ESWC on Emotions, Modality, Sentiment Analysis and the Semantic Web co-located with 14th ESWC 2017, Portroz, Slovenia, May 28, 2017. CEUR Workshop Proceedings 1874, CEUR-WS.org 2017

[19] Mauro Dragoni, Diego Reforgiato Recupero, Kerstin Denecke, Yihan Deng, Thierry Declerck: Joint Proceedings of the 2th Workshop on Emotions, Modality, Sentiment Analysis and the Semantic Web and the 1st International Workshop on Extraction and Processing of Rich Semantics from Medical Texts co-located with ESWC 2016, Heraklion, Greece, May 29, 2016. CEUR Workshop Proceedings 1613, CEUR-WS.org 2016

[20] Aldo Gangemi, Harith Alani, Malvina Nissim, Erik Cambria, Diego Reforgiato Recupero, Vitaveska Lanfranchi, Tomi Kauppinen: Joint Proceedings of the 1th Workshop on Semantic Sentiment Analysis (SSA2014), and the Workshop on Social Media and Linked Data for Emergency Response (SMILE 2014) co-located with 11th European Semantic Web Conference (ESWC 2014), Crete, Greece, May 25th, 2014. CEUR Workshop Proceedings 1329, CEUR-WS.org 2015

[21] Diego Reforgiato Recupero, Sergio Consoli, Aldo Gangemi, Andrea Giovanni Nuzzolese, Daria Spampinato: A Semantic Web Based Core Engine to Efficiently Perform Sentiment Analysis. ESWC (Satellite Events) 2014: 245-248

[22] Diego Reforgiato Recupero, Erik Cambria, Emanuele Di Rosa: Semantic Sentiment Analysis Challenge at ESWC2017. SemWebEval@ESWC 2017: 109-123

[23] Mauro Dragoni, Diego Reforgiato Recupero: Challenge on Fine-Grained Sentiment Analysis Within ESWC2016. SemWebEval@ESWC 2016: 79-94

[24] Diego Reforgiato Recupero, Mauro Dragoni, Valentina Presutti: ESWC 15 Challenge on Concept-Level Sentiment Analysis. SemWebEval@ESWC 2015: 211-222

[25] Keith Cortis, Andr Freitas, Tobias Daudert, Manuela Huerlimann, Manel Zarrouk, Siegfried Handschuh and Brian Davis. 2017. Semeval-2017 task 5:Fine-grained sentiment analysis on financial microblogs and news. In Proceedings of the 11th International Workshop on Semantic Evaluation . Association for Computational Linguistics, Vancouver, Canada, pages 517?533. (SemEval-2017)

[26] Thomas Gaillat, Manel Zarrouk, Andr Freitas and Brian Davis (2018). The SSIX Corpus: A Trilingual Gold Standard Corpus for Sentiment Analysis in Financial Microblogs.11th edition of the Language Resources and Evaluation Conference, 7-12 May 2018, Miyazaki (Japan). (LREC 2018)

[27] Amna Dridi, Mattia Atzeni, Diego Reforgiato Recupero: Bearish-Bullish Sentiment Analysis on Financial Microblogs. EMSASW@ESWC 2017

[28] Mattia Atzeni, Amna Dridi, Diego Reforgiato Recupero: Fine-Grained Sentiment Analysis on Financial Microblogs and News Headlines. SemWebEval@ESWC 2017: 124-128

Contents

Spanish Corpus of Tweets for Marketing

María Navas-Loro[1] (orcid.org/0000-0003-1011-5023), Víctor Rodríguez-Doncel[1] (orcid.org/0000-0003-1076-2511), Idafen Santana-Pérez[1] (orcid.org/0000-0001-8296-8629), Alba Fernández-Izquierdo[1], and Alberto Sánchez[2]

[1] Ontology Engineering Group, Universidad Politécnica de Madrid, Spain
[2] Havas Media, Madrid, Spain

Abstract. This paper presents a corpus of manually tagged tweets in Spanish language, of interest for marketing purposes. For every Twitter post, tags are provided to describe three different aspects of the text: the emotions, whether it makes a mention to an element of the marketing mix and the position of the tweet author with respect to the purchase funnel. The tags of every Twitter post are related to one single brand, which is also specified for every tweet. The corpus is published as a collection of RDF documents with links to external entities. Details on the used vocabulary and classification criteria are provided, as well as details on the annotation process.

Keywords: corpus, marketing, marketing mix, sentiment analysis, NLP, purchase funnel, emotion analysis

1 Introduction

Twitter is a source of valuable feedback for companies to probe the public perception of their brands. Whereas sentiment analysis has been extensively applied to social media messages (see [16] among many), other dimensions of brand perception are still of interest and have received less attention [12], specially those related to marketing. In particular, marketing specialists are highly interested in: (a) knowing the position of a tweet author in the purchase funnel (this is, where in the different stages of the customer journey is the author in); (b) knowing to which element or elements of the marketing mix[3] the text refers to and (c) knowing the author's affective situation with respect to a brand in the tweet.

This paper presents the MAS Corpus, a Spanish corpus of tweets of interest for marketing specialists, labeling messages in the three dimensions aforementioned. The corpus is freely available at `http://mascorpus.linkeddata.es/` and has been developed in the context of the Spanish research project LPS BIGGER[4], which analyzed different dimensions of tweets in order to extract relevant information on marketing purposes. A first version of the corpus containing only the sentiment analysis annotations was released as the Corpus for Sentiment

[3] `http://economictimes.indiatimes.com/definition/marketing-mix`
[4] `http://www.cienlpsbigger.es`

Analysis towards Brands (SAB) and was described in [15]. Following this work, we have expanded the corpus tagging the messages in the two remaining dimensions described before: the purchase funnel and the marketing mix. Tweets that were almost identical to others have been removed. Categories of each of the three aspects tagged in the corpus (Sentiment Analysis, Marketing Mix and Purchase Funnel) can be found in Table 1.

Table 1. Tags for each category.

Category	Tags
Purchase funnel	awareness, evaluation, purchase, postpurchase, ambiguous, NC2
Marketing Mix	product, price, promotion, place, NC2
Sentiment Analysis	love, hate, satisfaction, dissatisfaction, happiness, sadness, trust, fear, NC2

2 Related Work

2.1 Sentiment Analysis

Even when Sentiment Analysis is a major field in Natural Language Processing, most of works in Spanish tend to focus on polarity [10, 5], being the efforts towards emotions really scarce [22]. Sources of corpora also differ to our aims, since they tend to use specific websites or limit to domains such as tourism and medical opinions [17, 14] instead of social media. An extended review of works in Spanish Sentiment Analysis with regard to our needs can be found in [15].

2.2 Purchase Funnel

Although different purchase funnel interpretations have been suggested in literature [3, 6], we have based our approach on the one defined in the LPS BIGGER project and already used in [25]. This purchase funnel consists of four different stages (*Awareness, Evaluation, Purchase* and *Postpurchase*), that reflect how the client gets to know the product, investigates or compares it to other options, acquires it and actually uses and reviews it, respectively.

To the best of our knowledge, there are not public Spanish corpora available containing purchase funnel annotations, since the only work in Spanish on this topic the authors are aware of did not release the dataset used [25]. Nevertheless, the concept of Purchase Intention has been widely covered in literature, especially for marketing purposes in English language. Differently to Sentiment Analysis, Purchase Intention tries to detect or distinguish whether the client intends to buy a product, rather than whether he likes it or not [26]. Starting with the WISH corpus [8], covering wishes in several domains and sources (including product reviews), most works aim to discriminate between different kinds of intentions of users: in [21], the analysis focuses in suggestions and wishes for products and services both in a private dataset and in a part of the previously

mentioned WISH corpus; also an analysis performed on tweets about different intentions can be found in [13].

Finally, the most similar categories to the ones in our purchase funnel interpretation are the ones in [4], where the authors differentiate between several kinds of intention, being some of them (such as *wish*, *compare* or *complain*) easy mappable to our purchase funnel stages. Also the corpus used in [9], that classifies into pre-purchase and post-purchase reviews, shares our "timeline" interpretation of the purchase funnel. Out of the marketing domain, corpora labeled with purchase funnel tags for an specific domain have also been published, e.g., for the London musicals and recreational events [7].

2.3 Marketing Mix

Although the original concept of marketing mix [2] contained twelve elements for manufacturers, the most extended categorization for marketing is the one proposed by [11], consisting of four aspects (*price, product, promotion, place*) often known as "the four Ps" (or 4Ps) and revisited several times in literature [24]. Nevertheless, while marketing mix is a well-known and extended concept in the marketing field, in NLP the task of identifying these facets is often simply referred as detecting or recognizing "aspects", excepting some cases in literature [1]. This task has been often tackled in English [18, 20], while in Spanish corpora we can find a few datasets containing information about aspects, such as those in [5, 19].

3 Tagging Criteria

The corpus consists of more than 3k tweets of brands from different sectors, namely *Food, Automotive, Banking, Beverages, Sports, Retail* and *Telecom* (the complete list of brands, as well as statistics on the corpus, can be downloaded with it). When several brands appear in one tweet, just one of them is considered in the tagging process (the marked one); at the same time, the same tweet can appear several times in the corpus considering different brands. Every tweet is tagged in the dimensions exposed in Table 1; more than one tag is possible in sentiment and marketing mix dimensions (except simultaneously tagging the pairs of directly opposed emotions), while the purchase funnel, as representing a path on the purchase journey, only presents a tag per tweet. We describe below each dimension, along with a brief report on the criteria used for tagging each category (the complete criteria document can be downloaded with the corpus).

3.1 Sentiment Analysis

A tweet can be tagged with one or several emotions (as long as it does not contain directly opposite emotions), or with a *NC2* label meaning there are no emotions on it. Each basic emotion embraces also secondary emotions in it (described in Table 2), and a combination of them can express more complex feelings often seen in customers, such as shown in the following examples:

- When a customer is unable to find a desired product, the post is tagged as *sadness* (for the unavailability) and *satisfaction* (because it reveals previous satisfaction with the brand that deserves the effort of keep looking exactly for it instead of switching to one from another brand).
- When a post shows that a purchase is recurrent, it is tagged as *trust*, referring to the loyalty of the client.
- Emoticons of love are tagged as *love* and musical ones as *happiness* (unless irony happens). *Love* typically implies *happiness*.
- *Happiness* can only be tagged for an already acquired product or service.

Emotion	**Related emotions**
Trust	Optimism, Hope, Security
Satisfaction	Fulfillment, Contentment
Happiness	Joy, Gladness, Enjoyment, Delight, Amusement, Joviality, Enthusiasm, Jubilation, Pride, Triumph
Love	Passion, Excitement, Euphoria, Ecstasy
Fear	Nervousness, Alarm, Anxiety, Tenseness, Apprehension, Worry, Shock, Fright, Terror, Panic, Hysteria, Mortification
Dissatisfaction	Dislike, Rejection, Revulsion, Disgust, Irritation, Aggravation, Exasperation, Frustration, Annoyance
Sadness	Depression, Defeat, Hopelessness, Unhappiness, Anguish, Sorrow, Agony, Melancholy, Dejection, Loneliness, Humiliation, Shame, Guilt, Regret, Remorse, Disappointment, Alienation, Isolation, Insecurity
Hate	Rage, Fury, Wrath, Envy, Hostility, Ferocity, Bitterness, Resentment, Spite, Contempt, Vengefulness, Jealously

Table 2. Main emotions and their secondary emotions.

3.2 Purchase Funnel

Each tweet can belong to a stage in the purchase funnel, be ambiguous or be related to a brand without the author being involved in the purchase (such as is the case of posts of the brand itself). Different phases and concrete examples are tagged in the corpus as follows:

- **Awareness** The first contact of the client with the brand (either showing a willingness to buy or not), usually expressed in first person and mentioning advertising, videos, publicity campaigns, etc. Some examples of awareness would be:
  (1) *I just loved last Movistar ad.*
  (2) *I like the videos in Nike's YouTube channel.*

- **Evaluation** The post implies some research on the brand (such as questions or seek of confirmation) or comparison to others (by showing preferences among them, for instance), and some interest in acquiring a product or service. Examples of evaluation would be the following:
  (3) *I prefer Citroen to more expensive brands, such as Mercedes or BMW.*
  (4) *Looking for a second-hand Kia Sorento in NY, please send me a DM.*
- **Purchase** There is a direct reference to the moment of a purchase or to a clear intention of purchase (usually in first person). Some examples:
  (5) *I've finally decided to switch to Movistar.*
  (6) *Buying my brand new blue Citroen right now!*
- **Postpurchase** Texts referring to a past purchase or to a current experience, implying to own a product. This class presents a special complexity, since interpretation on the same linguistic patterns change depending on the kind of product, as already exposed in [25] and exemplified in the sentences below:
  (7) *I like Heineken, the taste is so good.*
  I would love a Heineken!
  (8) *I like BMWs, they are so classy!*
  I would love a BMW!
  In (7), the client has likely tasted that beer brand before; people does not tend to like or want beverages they have no experience with (at least without mentioning, such as in "*I want to taste the new Heineken.*"). But the same fact is not derived from more expensive items, even when expressed the same way, such as happens in (8): someone can like a car (such as its appearance or its engine) without having used it or intending to. This is why our criteria states that these kind of expressions must be tagged as *Postpurchase* for some brands (depending on the sector) and others must be tagged as *Ambiguous*, since there can be several possible and equally likely interpretations.
- **Ambiguous** This category includes critical posts, suggestions and recommendations, along with posts where it is not clear in which stage the customer is (such as the case mentioned above).
  (9) *Do not buy Milka!*
  (10) *Loving the new Kia!*
- **NC2** Includes impersonal messages without opinions (such as corporative news or responses of the brand to clients), questions implying no personal evaluation or intention (for instance, involving a third person), texts with buy or rental offers with no mention to real use experience, etc.
  (11) *2008 Hyundai for sale.*
  (12) *My aunt didn't like the Kia.*

3.3 Marketing Mix

We have added a *NC2* class to the four original McCarthy's Ps to indicate none of the four aspects is treated in the tweet. It must be noted that, differently than the purchase funnel, several marketing mix tags can appear in the same tweet (except of the *NC2*). Brief explanation of each of the categories tagged for marketing mix, along with examples and part of the criteria, are exposed below:

- **Product** This category encompasses texts related to the features of the product (such as its quality, performance or taste), along with references to design (such as size, colors or packing) or guaranty, such as in the following examples:
  (13) *I find the new iPhone too big for my pocket.*
  (14) *I love the new mix Milka Oreo!*
  Note that when someone loves/likes something (such as food), we assume it refers to some feature of a product (such as its taste), so we tag it as *Product*.
- **Promotion** Texts referring to all the promotions and programs of the brand channeled to increase sales and ensure visibility to their products or the brand, such as advertisements, sponsorships (such as prices, sport teams or events), special offers, work offers, promotional articles, etc.
  (15) *Freaking out with the new 2x1 @Ikea!*
  (16) *La Liga BBVA is the best league in the world.*
- **Price** Includes economical aspects of a product, such as references to its value or promotions involving discounts or price drops (that must also be tagged as *Promotion*). Examples of texts that should be tagged as *Price* would be the following:
  (17) *I'm afraid that I can't afford the new Toyota.*
  (18) *Yesterday I saw the same Adidas for just 40e!*
- **Place** Aspects related to commercialization, such physical places of distribution of the products (for instance, if a product is difficult to find) and customer service (in every stage of the purchase: information, at the point of sale, postpurchase, technical support, etc).
  (19) *I love the new Milka McFlurry at McDonalds*
  (20) *Already three malls and unable to find the new Nike Pegasus!*
- **NC2** Impersonal messages of the brand, news or texts that include none of the aspects mentioned before.
  (21) *Nike is paying no tax!*
  (22) *I can't decide between Puleva and Pascual.*

4 The MAS Corpus

4.1 Building the corpus

A different approach was used for Marketing Mix and the Purchase Funnel tagging with respect to the Sentiment Analysis tagging procedure (where three taggers acted independently with just a common criteria document) exposed in [15]. This meets the need of streamlining the whole tagging process, that happens to be both difficult and time-consuming for taggers. This new procedure is briefly exposed below:

1. A first version of the criteria document was written, based on the study of literature and previous experience within the LPS BIGGER project.

2. Then Tagger 1 tagged a representative part of the corpus (about 800 tweets), highlighting main doubts and dubious tweets with regard to the criteria, that are revised; new tagging examples are added, and some nuances and special cases are rewritten.
3. Taggers 2 and 3 revise the tags by Tagger 1, paying special attention to tweets marked as dubious: if an agreement is reached, the tagging is updated consequently; otherwise, the tweet is tagged as *Ambiguous* or *NC2*.
4. Then each tagger takes a part of the corpus to tag it following the new criteria and highlighting doubts again; these tweets will be revised with remaining taggers, reaching an agreement on the final unique tags in the corpus.

4.2 Publishing the corpus as Linked Data

We maintain the RDF representation used in the previous version of the corpus, using again our own vocabulary[5] to express the purchase funnel and the marketing mix. We also reuse Marl [27] and Onyx [23] for emotions and polarity, and SIOC[6] and GoodRelations[7] for post and brand representation. Also links to the entries of brands and companies in external databases such as Thomson Reuters' PermID[8] and DBpedia[9] extend the information in the tweets. Fig. 1 shows an example of a tweet tagged in the dimensions extracted from the corpus.

4.3 Corpus description

Final corpus contains 3763 tweets. Statistics on linguistic information in the corpus can be found in Table 3, along with specific data relevant for Social Media, such as the amount of hashtags, user mentions and URLs. The distribution of categories varies depending on the sector, as shown in Table 4. Mentions of *Place* are for instance more common in *Sports* than in other categories, such as *Beverages* or *Telecom*. Also when opinions are expressed differs: tweets in the *Food* sector tend to refer to the *Postpurchase* phase, while others tend to be more ambiguous or refer to previous phases. Regarding emotions, some of them just appear in certain domains, such as *Fear* for *Banking*.

5 Conclusions

Whereas the SAB corpus provided a collection of tweets tagged with labels useful for making Sentiment Analysis towards brands, this new corpus is of interest for the marketing analysis in a broader way; the MAS Corpus allows marketing professionals to have additional information of habits and behaviors, strong and weak points of the whole purchase experience, and also full insights on concrete aspects of each client reviews.

[5] `http://sabcorpus.linkeddata.es/vocab`
[6] `https://www.w3.org/Submission/sioc-spec/`
[7] `http://purl.org/goodrelations/`
[8] `https://permid.org/`
[9] `http://dbpedia.org/`

Table 3. Total and average (per tweet) statistics on the corpus. Stanford CoreNLP was used for POS information, while patterns were used for detecting hashtags (*'#'*), mentions (*'@'*) and URLs (*'www.*'/'http*'*).

	TOTAL	AVG		TOTAL	AVG
Tweets	3763	-	**Verbs**	6971	1.85
Sentences	5189	1.38	**Nouns**	8353	2.22
Tokens	59555	15.83	**NPs**	6952	1.85
Hashtags	1819	0.48	**Adjectives**	2761	0.73
Mentions	2306	0.61	**Adverbs**	1584	0.42
URLs	2111	0.56	**Neg. Adverbs**	560	0.15

Table 4. Statistics on the corpus. Column *ANY* in emotional categories shows the percentage of posts with any emotion (this is, non neutral posts); remaining columns show the percentage of each category among these non neutral posts. For Purchse Funnel and Marketing Mix, each column represents the percentages of each of the tags described in Section 3.

	ANY	HAT	SAD	FEA	DIS	SAT	TRU	HAP	LOV
FOOD	54.79	1.50	1.20	0.00	8.08	45.21	44.01	14.67	12.87
AUTOMOTIVE	9.11	0.00	0.22	1.11	2.44	6.89	3.33	1.11	0.89
BANKING	24.67	5.33	1.00	15.00	23.83	1.33	0.50	0.00	0.00
BEVERAGES	63.11	2.07	1.19	0.74	19.11	44.00	32.74	7.26	7.70
SPORTS	34.15	2.45	2.60	0.31	13.32	18.84	11.94	4.90	11.33
RETAIL	33.00	3.20	1.11	1.48	11.95	14.53	14.41	3.69	3.45
TELECOM	40.17	12.97	0.84	0.00	30.13	8.79	6.28	3.35	1.26

	PURCHASE FUNNEL						MARKETING MIX				
	NC2	AWA	EVA	PUR	POS	AMB	NC2	PROD	PRI	PROM	PLA
FOOD	43.41	3.59	3.29	4.19	40.72	5.09	48.80	30.84	2.10	15.27	7.49
AUTOMOTIVE	85.56	2.67	4.00	0.22	4.44	3.33	77.56	4.67	2.00	16.00	1.56
BANKING	58.50	5.83	2.00	0.00	7.83	25.67	53.33	8.50	7.83	21.17	13.17
BEVERAGES	33.63	0.44	13.33	8.44	11.26	32.74	19.85	70.37	2.22	8.59	8.59
SPORTS	63.09	2.91	4.29	1.84	7.50	19.75	54.98	6.43	17.76	0.92	30.32
RETAIL	89.29	2.71	4.80	0.62	1.97	1.60	72.17	12.56	2.09	8.62	7.51
TELECOM	94.14	0.42	0.42	0.00	4.60	0.00	91.63	1.26	1.67	4.60	0.00

```
mas:827146264517165056 a sioc:Post ;
  sioc:id "827146264517165056" ;
  sioc:content "Las camisetas nike 2002~2004 y las adidas 2006~2008 son el amor de mi vida"@es ;
  marl:describesObject mas:Nike ;
  sabd:isInPurchaseFunnel sabv:postPurchase;
  sabd:hasMarketingMix sabv:product;
  onyx:hasEmotion sabv:love, sabv:satisfaction, sabv:happiness ;
  marl:hasPolarity marl:positive ;
  marl:forDomain "SPORT" .

  mas:Nike a gr:Brand ;
  rdfs:seeAlso <http://dbpedia.org/resource/Nike> ;
  sabd:1-5000062703 a gr:Business ;
  rdfs:label "Nike Inc", "Nike" ;
  owl:sameAs permid:1-4295904620 .
```

Fig. 1. Sample tagged post, and extra information on its brand (Nike) and company (Nike Inc).

Acknowledgments. This work has been partially supported by LPS-BIGGER (IDI-20141259), esTextAnalytics project (RTC-2016-4952-7), Datos 4.0 project with ref. TIN2016-78011-C4-1-R, a Predoctoral grant by the Consejo de Educación, Juventud y Deporte de la Comunidad de Madrid partially founded by the European Social Fund, two Predoctoral grants from the I+D+i program of the Universidad Politécnica de Madrid and a Juan de la Cierva contract. We would also want to thank Pablo Calleja for his help in corpora statistics extraction.

References

1. Bel, N., Diz-pico, J., Pocostales, J.: Classifying short texts for a Social Media monitoring system Clasificación de textos cortos para un sistema monitor de los Social Media. Procesamiento del Lenguaje Natural 59, 57–64 (2017)
2. Borden, N.H.: The concept of the marketing mix. Journal of advertising research 4(2), 2–7 (1964)
3. Bruyn, A.D., Lilien, G.L.: A multi-stage model of word-of-mouth influence through viral marketing. Int. Journal of Research in Marketing 25(3), 151–163 (2008)
4. Cohan-Sujay, C., Madhulika, Y.: Intention Analysis for Sales, Marketing and Customer Service. Proceedings of COLING 2012, Demonstration Papers, (December 2012), 33–40 (2012)
5. Cumbreras, M.Á.G., Cámara, E.M., et al.: TASS 2015 - The evolution of the Spanish opinion mining systems. Procesamiento de Lenguaje Natural 56, 33–40 (2016)
6. Elzinga, D., Mulder, S., Vetvik, O.J., et al.: The consumer decision journey. McKinsey Quarterly 3, 96–107 (2009)
7. García-Silva, A., Rodríguez-Doncel, V., Corcho, Ó.: Semantic characterization of tweets using topic models: A use case in the entertainment domain. Int. J. Semantic Web Inf. Syst. 9(3), 1–13 (2013)
8. Goldberg, A.B., Fillmore, N., Andrzejewski, D., Xu, Z., Gibson, B., Zhu, X.: May All Your Wishes Come True : A Study of Wishes and How to Recognize Them. Proceedings of Human Language Technologies: NAACL '09 (June), 263–271 (2009)
9. Hasan, M., Kotov, A., Mohan, A., Lu, S., Stieg, P.M.: Feedback or Research: Separating Pre-purchase from Post-purchase Consumer Reviews. In: Advances in

Information Retrieval. ECIR 2016. Lecture Notes in Computer Science, vol. 9626, pp. 682–688. Springer, Cham (2016)
10. Martínez-Cámara, E., Martín-Valdivia, M.T., et al.: Polarity classification for Spanish tweets using the COST corpus. Journal of Information Science 41(3), 263–272 (jun 2015)
11. McCarthy, E.: Basic Marketing, a Managerial Approach. Sixth Edition, Homewood, Ill.: Richard D. Irwin, Inc. (1978)
12. Moghaddam, S.: Beyond sentiment analysis: Mining defects and improvements from customer feedback. LNCS 9022, 400–410 (2015)
13. Mohamed, H., Mohamed, S.G., Lamjed, B.S.: Customer Intentions Analysis of Twitter Based on Semantic Patterns. 2015 pp. 2–6 (2015)
14. Molina-González, M.D., Martínez-Cámara, E., et al.: Cross-domain sentiment analysis using Spanish opinionated words. In: Proceedings of NLDB. pp. 214–219 (2014)
15. Navas-Loro, M., Rodríguez-Doncel, V., Santana-Perez, I., Sánchez, A.: Spanish Corpus for Sentiment Analysis towards Brands. In: Proc. of the 19th Int. Conf. on Speech and Computer (SPECOM). pp. 680–689 (2017)
16. Pak, A., Paroubek, P.: Twitter as a corpus for sentiment analysis and opinion mining. In: LREc. vol. 10 (2010)
17. Plaza-Del-Arco, F.M., Martín-Valdivia, M.T., et al.: COPOS: Corpus of patient opinions in Spanish. Application of sentiment analysis techniques. Procesamiento de Lenguaje Natural 57, 83–90 (2016)
18. Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., Manandhar, S.: Semeval-2014 task 4: Aspect based sentiment analysis pp. 27–35 (01 2014)
19. Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., AL-Smadi, M., Al-Ayyoub, M., Zhao, Y., Qin, B., De Clercq, O., Hoste, V., Apidianaki, M., Tannier, X., Loukachevitch, N., Kotelnikov, E., Bel, N., Jiménez-Zafra, S.M., Eryiğit, G.: Semeval-2016 task 5: Aspect based sentiment analysis. In: Proceedings SemEval-2016. pp. 19–30. ACL, San Diego, California (June 2016)
20. Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., AL-Smadi, M., Al-Ayyoub, M., Zhao, Y., Qin, B., De Clercq, O., et al.: Semeval-2016 task 5: Aspect based sentiment analysis. In: ProWorkshop SemEval-2016. pp. 19–30. ACL (2016)
21. Ramanand, J., Bhavsar, K., Pedanekar, N.: Wishful thinking: finding suggestions and'buy'wishes from product reviews. Proceedings of the NAACL HLT 2010 Workshop CAAGET '10) (June), 54–61 (2010)
22. Rangel, F., Rosso, P., Reyes, A.: Emotions and Irony per Gender in Facebook. In: Proceedings of Workshop ES3LOD, LREC-2014,. pp. 1–6 (2014)
23. Sánchez Rada, J.F., Torres, M., et al.: A linked data approach to sentiment and emotion analysis of twitter in the financial domain. In: FEOSW (2014)
24. Van Waterschoot, W., Van den Bulte, C.: The 4p classification of the marketing mix revisited. The Journal of Marketing pp. 83–93 (1992)
25. Vázquez, S., Muñoz-García, O., Campanella, I., Poch, M., Fisas, B., Bel, N., Andreu, G.: A classification of user-generated content into consumer decision journey stages. Neural Networks 58(Supplement C), 68–81 (2014), Special Issue on "Affective Neural Networks and Cognitive Learning Systems for Big Data Analysis"
26. Vineet, G., Devesh, V., Harsh, J., Deepam, K., Shweta, K.: Identifying purchase intent from social posts. ICWSM 2014 pp. 180–186 (2014)
27. Westerski, A., Iglesias, C.A., Rico, F.T.: Linked opinions: Describing sentiments on the structured web of data. In: Proceedings of the 4th International Workshop Social Data on the Web. vol. 830 (2011)

Supervised Topic-Based Message Polarity Classification using Cognitive Computing

Daniele Stefano Ferru, Federico Ibba, and Diego Reforgiato Recupero

Department of Mathematics and Computer Science
University of Cagliari, Italy
d.s.ferru@outlook.it, federico.ibba@unica.it, diego.reforgiato@unica.it

Abstract. This paper describes a supervised approach we have designed for the topic-based message polarity classification. Given a message and a topic, we aim at (i) classifying the message on a two point scale, that is positive or negative sentiment toward that topic and (ii) classifying the message on a five-point scale, that is the message conveyed by that tweet toward the topic on a more fine-grained range. These two tasks have been proposed as subtasks of SemEval-2017 task 4. We have targeted them with the employment of IBM Watson that we leveraged to extract concepts and categories to enrich the vectorial space we have modeled to train our classifiers. We have used different classifiers for the two tasks on the provided training set and obtained good accuracy and F1-score values comparable to the SemEval 2017 competitors of those tasks.

Keywords: Sentiment Analysis, NLP, Polarity Detection, Cognitive Computation, Linear Regression, Decision Tree, Naive Bayes

1 Introduction

Social media platforms are commonly used to share opinions and thoughts about different subjects and topics in any domain. Their huge widespread and proliferation of content has created opportunities to analyze and study opinions, how and where emotions are generated, what the current feelings are on a certain topic and so on. It is straightforward therefore to understand that social media have more and more interest in identifying sentiment in document, messages or posts. The common task is to detect whether in a given text there are positive, negative, neutral opinions expressed, and whether these opinions are general or focused on a certain person, product, organization or event. A lot of research has been already performed to address this task and several variations and extensions of it [3, 13]. On the one hand, supervised and unsupervised approaches have been proposed based on Natural Language Processing (NLP) techniques, machine learning tools, statistics. On the other hand, semantics has already shown to provide benefits to supervised approaches for Sentiment Analysis [26, 10, 21] where extracted semantic features enrich the vectorial space to be fed to machine learning tools (classifiers) through augmentation, replacement and

interpolation techniques leading to higher accuracy. Semantics has been leveraged in unsupervised approaches too for Sentiment Analysis: authors in [24, 14] have introduced Sentilo, a sentic computing approach to opinion mining that produces a formal representation (e.g. a RDF graph) of an opinion sentence that allows distinguishing its holders and topics with very high accuracy. They have also defined and extended an ontology for opinion sentences, created a new lexical resources enabling the evaluation of opinion expressed by means of events and situations and developed an algorithm to propagate the sentiment towards the targeted entities in a sentence.

Cognitive computation is a recent kind of technology that is specialized in the processing and analysis of large unstructured datasets by leveraging artificial intelligence, signal processing, reasoning, NLP, speech recognition and vision, human-computer interaction, dialog and narrative generation. Cognitive computing systems have earned a lot of attention for figuring out relevant insights from textual data such as classifying biomedical documents [5] and e-learning videos [4]. One of the most known systems is IBM Watson[1] which can understand concepts, entities, sentiments, keywords, etc. from unstructured text through its Natural Language Understanding[2] service.

In this paper we propose a supervised approach for topic-based message polarity classification formulated as follows: given a message and a topic, classify the message on a two-point scale (Task 1) and on a five-point scale (Task 2). These two tasks have been proposed within the task 4: Sentiment Analysis in Twitter of SemEval 2016 [19][3] and SemEval 2017 [25][4].

We used machine learning approaches to target the two tasks above and leveraged IBM Watson to extract concepts and categories from the input text and to augment the vectorial space using term frequency and TF-IDF. Training and test data consist of tweets and a given topic for each tweet. As for each topic we have several tweets, we created as many classifiers as the overall number of topics in the training set. During the prediction step for a given pair (tweet, topic), two possibilities might occur:

1. the topic was found within the training set and therefore we selected the classifier already trained on the tweets related to that topic;
2. The topic was not found in any tweets of the training set. To solve this case, we used the classifier on the closest topic to the one to predict. We leveraged the semantic features extracted by IBM Watson to find the closest topic in the training set to the one to predict.

The performance evaluation we have carried out indicates satisfying results for the Task 1 whereas for Task 2 they suffer from the low number of tweets per topic present within the training set with respect to the number of tweets in the test set.

[1] https://www.ibm.com/watson/
[2] https://www.ibm.com/watson/services/natural-language-understanding/
[3] http://alt.qcri.org/semeval2016/task4/
[4] http://alt.qcri.org/semeval2017/task4/

The remainder of this paper is organized as follows. Section 2 describes background work on Sentiment Analysis techniques and how Semantics has been employed in that domain. Section 3 introduces the data we have used and how they are organized. Section 4 includes details on the method we have adopted to tackle the tasks and how Cognitive Computing has been leveraged. Section 5 shows results we have obtained and the evaluation we have carried out. Section 6 depicts concluding remarks.

2 Related Work

Several initiatives (challenges [22, 6, 23], workshop, conferences) within the Sentiment Analysis domain have been proposed. As mentioned in Section 1, the tasks we are targeting in this paper have been proposed by SemEval 2016 and SemEval 2017 task 4 where SemEval is an ongoing series of evaluations of computational semantic analysis systems, organized under the umbrella of SIGLEX, the Special Interest Group on the Lexicon of the Association for Computational Linguistics.

Authors in [28] investigated a method based on Conditional Random Fields to incorporate sentence structure (syntax and semantic) and context information to detect sentiments. They have also employed the Rethorical Structure Theory leveraging the discourse role of text segments and proved the effectiveness of the two features on the Movie Review Dataset and the Fine-grained Sentiment Dataset. Within the financial domain, authors in [9] proposed a fine-grained approach to predict real valued sentiment score by using feature sets consisting of lexical features, semantic features and their combination. Multi-domain sentiment analysis has been further targeted by authors in [7, 8] that suggested different general approaches using different features such as word embeddings. Semantic features can be extracted by several lexical and semantic resources and ontologies. Today, with the recent widespread of cognitive computing tools, we have one more tool we can leverage to refine our extraction. Cognitive computing systems [15, 16] are in fact emerging tools and represent the third era of computing. They have been used to improve not only the sentiment analysis [24], but also multi-class classification of e-learning videos [4], classification of complaints in the insurance industry [12] and within life sciences research [2]. These systems rely on deep learning algorithms and neural networks to elaborate information by learning from a training set of data. They are perfectly tailored to integrate and analyze the huge amount of data that is being released and available today. Two very well known cognitive computing systems are IBM Watson[5] and Microsoft Cognitive Services[6]. In this paper we have leveraged the former to extract categories and concepts out of an input tweet. Many others articles are presented every year within the Sentiment Analysis domain, and, therefore, several survey papers have been drafted to summarize the recent research trends and directions [27, 17, 20, 1, 11, 18].

[5] https://www.ibm.com/watson/
[6] https://azure.microsoft.com/en-us/services/cognitive-services/

3 The Used Dataset

The data have been obtained from SemEval[7]. The have been extracted from Twitter and annotated using CrowdFlower[8]. The datasets (training and test) for Task 1 included a tweet id, the topic, the tweet text and the tweet classification as positive, negative and neutral. The datasets for Task 2 (training and test) had the same structure except for the tweet classification that was an integer number ranging in [-2, +2]. Tables 1 and 2 show, respectively, five records of the dataset related to Task 1 and Task 2.

Table 1. Sample tweets for Task 1.

Tweet Id	**Topic**	**Tweet class**	**Tweet text**
522712800595300352	aaron rodgers	neutral	I just cut a 25 second audio clip of Aaron Rodgers talking about Jordy . Nelson's grandma's pies. Happy Thursday.
523065089977757696	aaron rodgers	negative	@Espngreeny I'm a Fins fan, it's Friday, and Aaron Rodgers is still giving me nightmares 5 days later. I wished it was a blowout.
522477110049644545	aaron rodgers	positive	Aaron Rodgers is really catching shit for the fake spike Sunday night.. Wtf. It worked like magic. People just wanna complain about the L.
522551832476790784	aaron rodgers	neutral	If you think the Browns should or will trade Manziel you're an idiot. Aaron Rodgers sat behind Favre for multiple years.
522887492333084674	aaron rodgers	neutral	Green Bay Packers: Five keys to defeating the Panthers in week seven: Aaron Rodgers On , Sunday ... http://t.co/anCHQjSLh9 #NFL #Packers

Moreover, Table 3 indicates the size of training sets and test sets for the two tasks whereas Table 4 and Table 5 show some statistics of the data.

4 The Proposed Method

In order to prepare the vectorial space, we have augmented the bag of words model resulting from the tweets of the training set with two kind of semantic features extracted using IBM Watson: categories and concepts. As an example, for the third tweet of Table 1 IBM Watson has extracted as categories *magic and illusion, football, podcasts* and as concepts *2009, singles*.

We have employed the augmentation method mentioned in [10] to create different vectorial spaces that we have adopted to evaluate the performances of

[7] http://alt.qcri.org/semeval2017/task4/index.php?id=data-and-tools
[8] https://www.crowdflower.com/

Table 2. Sample tweets for Task 2.

Tweet Id	Topic	Tweet class	Tweet text
681563394940473347	amy schumer	-1	@MargaretsBelly Amy Schumer is the stereotypical 1st world Laci Green feminazi. Plus she's unfunny
675847244747177984	amy schumer	-1	dani_pitter I mean I get the hype around JLaw. I may not like her but I get her hype. I just don't understand Amy Schumer and her hype
672827854279843840	amy schumer	-1	Amy Schumer at the #GQmenoftheyear2015 party in a dress we pretty much hate: https://t.co/j5HmmyM99j #GQMOTY2015 https://t.co/V8xzmPmPYX
662755012129529858	amy schumer	-2	Amy Schumer is on Sky Atlantic doing one of the worst stand up sets I have ever seen. And I've almost sat through 30 seconds of Millican.
679507103346601984	amy schumer	2	"in them to do it. Amy Schumer in EW, October amyschumer is a fucking rock star & I love her & Jesus F'ing Christ we need more like this" #NFL #Packers

Table 3. Sizes of the training and test sets for the two targeted tasks.

	Training Set	Test Set
#Task 1	16496	4908
#Task 2	23776	11811

Table 4. Statistics of the training and test sets for Task 1.

	# of Pos Tweets	# of Neg Tweets	# of Neutral Tweets
Training Set	9852	5649	995
Test Set	3780	914	214

Table 5. Distribution of the five classes for the training and test sets of Task 2.

	# Class -2	# Class -1	# Class 0	# Class 1	# Class 2
Training Set	210	2563	10216	10016	771
Test Set	172	3377	5871	2261	130

our methods. In particular we have employed the vectorial space consisting of: (i) tweets only (what we refer as baseline), (ii) tweets augmented with categories, (iii) tweets augmented with concepts, (iv) and tweets augmented with categories and concepts. We performed a set of cleaning steps to the resulting bag of words which included (i) lower casing the tokens of the input tweets, categories and concepts, (ii) removing of special characters and numbers, (iii) removing of stop words taken from StanfordNLP[9].

We employed machine learning classifiers and fed them with the produced vectorial spaces. In particular we used Linear Regression and Naive Bayes for the binary prediction of Task 1 where we have considered the positive/negative classes getting rid of the neutral class (as also suggested in the corresponding SemEval task). As far as the multi class classification of the Task 2 is concerned, we employed Decision Trees and Naive Bayes classifiers. To note that, because our data consisted of a set of tweets for each topic, we have trained a classifier for each topic in the training set feeding it with all the tweets with that topic. Both the tasks we targeted are topic-based and, therefore, given a tweet and a topic, we first had to find the most similar topic in the training set and then use the related classifier for the prediction step.

4.1 Associating Test Set and Training Set topics

Since the topics in the test set are completely different from those in the training set, we had to choose a strategy to associate the most similar topic of the training set (and therefore pick the related classifier) with each topic in the test set. To achieve this we used the categories obtained by IBM Watson. Every tweet in the training set has different related categories, thus a set with all the categories for each topic has been prepared. Similarly, for each topic in the test set, we prepared a set of all the categories extracted from each tweet related to that topic. Therefore, each topic in the training set and in the test set corresponded to a vector of categories. During the prediction of a given tweet with a certain topic t, we needed to use the classifier trained on the tweets having the most similar topic to t. To find the most similar topic in the training set to t, we counted how many categories the two lists (one corresponding to t and the other corresponding to each topic in the training set) had in common and took the one with the highest number.

5 Performance Evaluation

According to SemEval, the evaluation measure for Task 1 was the average recall that we refer as *AvgRec*:

$$AvgRec = \frac{1}{2} \cdot (R^P + R^N)$$

[9] https://bit.ly/1Nt4eMh

where R^P and R^N refer to the recall with respect to the positive and negative class. $AvgRec$ ranges in [0,1] where a value of 1 is obtained only by a perfect classification and 0 is obtained in presence of a classifier that misclassifies all the items. The F1 score has further been used as secondary measure for Task 1. It is computed as:

$$F1 = 2 \cdot \frac{(P^P + P^N) \cdot (R^P + R^N)}{P^P + P^N + R^P + R^N}$$

As the task is topic-based we have computed each metric individually for each topic and then we computed the average value across all the topics to obtain the final score. Task 2 is a classification where we need to classify a tweet in exactly one class among those defined in C={highly negative, negative, neutral, positive, highly positive} represented in our data by {-2, -1, 0, 1, 2}. We used macro-average mean absolute error (MAE^M) defined as:

$$MAE^M(h, Te) = \frac{1}{|C|} \cdot \sum_{j=1}^{|C|} \frac{1}{|Te_j|} \cdot \sum_{x_i \in Te_j} |h(x_i) - y_i|$$

where y_i denotes the true label of item x_i, $h(x_i)$ is its prediction, Te_j represents the set of test documents having c_j as true class, $|h(x_i) - y_i|$ is the distance between classes $h(x_i)$ and y_i.

One benefit of the MAE^M measure is that it is able to recognize major misclassifications: for example misclassifying a highly negative tweet in highly positive is worse than misclassifying it as negative. We also used the standard mean absolute error MAE^μ, which is defined as:

$$MAE^\mu(h, Te) = \frac{1}{|Te|} \cdot \sum_{x_i \in Te} |h(x_i) - y_i|$$

The advantage of MAE^M with respect to MAE^μ is that it is robust to unbalanced class (as in our case) whereas the two measures are equivalent in presence of balanced datasets. Both MAE^M and MAE^μ have been computed for each topic and results averaged across all the topics to obtain one final score.

Tables 6 and 7 show the results we obtained for our proposed Task 1 whereas Tables 8 and 9 include results for Task 2. Results for both the tasks have been obtained by using the training and test sets of the data released from SemEval and also using a 10-cross validation by merging them. In the latter case, we did not consider the topic information during the learning step and trained one single classifier that used for the test.

5.1 Discussion of the results

In this section we discuss the obtained results for the two tasks we targeted in this paper. On the one hand, the employment of the semantic features had an impact for the classification within Task 1. As the Tables 6 and 7 show, adding the categories to the baseline improved the overall results. The addition

Table 6. Results of AvgRec and F1 values for Task 1 using the test set of SemEval.

	Baseline	**Tweets+Ctg**	**Tweets+Conc**	**Tweets+Ctg+Conc**
AvgRec				
Linear Regression	0.4438	0.4942	0.4515	0.4982
Naive Bayes	0.4628	0.4946	0.4604	0.4969
F1-value				
Linear Regression	0.5566	0.6339	0.5856	0.6316
Naive Bayes	0.5159	0.5200	0.5052	0.5104

Table 7. Results of AvgRec and F1 values for Task 1 using 10-cross validation on the union of training and test sets.

	Baseline	**Tweets+Ctg**	**Tweets+Conc**	**Tweets+Ctg+Conc**
AvgRec				
Linear Regression	0.485	0.505	0.484	0.506
Naive Bayes	0.492	0.522	0.493	0.522
F1-value				
Linear Regression	0.649	0.654	0.647	0.651
Naive Bayes	0.619	0.606	0.613	0.603

Table 8. Results of MAE^M and MAE^μ for Task 2 using the test set of SemEval.

	Baseline	**Tweets+Ctg**	**Tweets+Conc**	**Tweets+Ctg+Conc**
MAE^M				
Decision Trees	3.628	4.207	3.745	4.242
Naive Bayes	9.548	12.02	9.882	12.34
MAE^μ				
Decision Trees	0.472	0.552	0.488	0.559
Naive Bayes	1.219	1.556	1.256	1.601

Table 9. Results of MAE^M and MAE^μ for Task 2 using 10-cross validation on the union of training and test sets.

	Baseline	**Tweets+Ctg**	**Tweets+Conc**	**Tweets+Ctg+Conc**
MAE^M				
Decision Trees	1.292	1.317	1.299	1.320
Naive Bayes	1.930	2.196	1.984	2.250
MAE^μ				
Decision Trees	0.586	0.603	0.586	0.605
Naive Bayes	1.058	1.196	1.085	1.221

of concepts only does not help the classification process as with the categories probably because the lower number of concepts ends up adding noise in the used classifiers (Naive Bayes and Linear Regression). Results are confirmed also with the 10-cross-validation.

On the other hand, Task 2 shows important differences between the baseline and the tweets with the semantic features as Task 1 but in the opposite direction. As Tables 8 and 9 show, adding semantic features never improves the classification results, indicating they act like noise. This might be justified given the unbalanced nature of the used dataset: typically, each topic contains more tweets for a few classes and much less for the others. This fact generate a lot of error in the classification task and produces poor results. Furthermore, one explanation of such a behaviour is that Task 1 only consisted of a binary classification whereas Task 2 consisted of the multiclass classification where the output class might be assigned to one of five different values. Predicting five values instead of two is much harder and, given the low number of tweets per topic, the classifiers could not be trained well enough on an appropriate dataset.

6 Conclusion

In this paper we have presented a supervised topic-based message polarity classification for two tasks proposed at SemEval. The first task aims at classifying a tweet on a two point scale (positive or negative) toward a given topic. The second task aims at classifying a tweet on a five-point scale. We have targeted the two tasks using a machine learning approach where the vectorial space has been created by augmenting the message (tweets) with semantic features (categories and concepts) extracted with IBM Watson, a well known cognitive computing tool. Moreover, categories and concepts have been used to calculate the distances between topics of the training set and test set in order to associate the latter to the former. Although the low number of tweets in the training set, for Task 1 we obtained good results whereas Task 2 suffered from the scarcity of training data. Obtained results showed that with few classes (Task 1), concepts and categories were important for the classification task. Conversely, given the strong unbalanced nature of the dataset, in Task 2 concepts and categories were not able to enrich the obtained vectorial space. To address this issue, and as next steps, we would like to further investigate the employment of semantic features extracted from other cognitive computing systems trying to combine and compare them with the results obtained using IBM Watson.

Acknowledgments

The authors gratefully acknowledge Sardinia Regional Government for the financial support (Convenzione triennale tra la Fondazione di Sardegna e gli Atenei Sardi Regione Sardegna - L.R. 7/2007 annualità 2016 - DGR 28/21 del 17.05.201, CUP: F72F16003030002).

References

1. E. Cambria, B. Schuller, Y. Xia, and C. Havasi. New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems*, 28(2):15–21, March 2013.
2. Ying Chen, JD Elenee Argentinis, and Griff Weber. Ibm watson: How cognitive computing can be applied to big data challenges in life sciences research. *Clinical Therapeutics*, 38(4):688 – 701, 2016.
3. Keith Cortis, Andre Freitas, Tobias Daudert, Manuela Huerlimann, Manel Zarrouk, Siegfried Handschuh, and Brian Davis. Semeval-2017 task 5: Fine-grained sentiment analysis on financial microblogs and news. *Proceedings of the 11th International Workshop on Semantic Evaluation*, pages 517–533, 2017.
4. Danilo Dessì, Gianni Fenu, Mirko Marras, and Diego Reforgiato Recupero. Leveraging cognitive computing for multi-class classification of e-learning videos. In *The Semantic Web: ESWC 2017 Satellite Events - ESWC, Portorož, Slovenia, May 28 - June 1, 2017, Revised Selected Papers*, pages 21–25, 2017.
5. Danilo Dessì, Diego Reforgiato Recupero, Gianni Fenu, and Sergio Consoli. Exploiting cognitive computing and frame semantic features for biomedical document clustering. In *Proc. of the Workshop on Semantic Web Solutions for Large-scale Biomedical Data Analytics co-located with 14th Extended Semantic Web Conference, Portoroz, Slovenia, May 28, 2017.*, pages 20–34, 2017.
6. M. Dragoni and D. Reforgiato Recupero. Challenge on fine-grained sentiment analysis within eswc2016. *Communications in Computer and Information Science*, 641:79–94, 2016.
7. Mauro Dragoni and Giulio Petrucci. A neural word embeddings approach for multi-domain sentiment analysis. *IEEE Trans. Affective Computing 8(4)*, pages 457–470, 2017.
8. Mauro Dragoni and Giulio Petrucci. A fuzzy-based strategy for multi-domain sentiment analysis. *International Journal of Approximate Reasoning*, 93:59–73, 2018.
9. Amna Dridi, Mattia Atzeni, and Diego Reforgiato Recupero. Bearish-bullish sentiment analysis on financial microblogs. In *Proc. of EMSASW 2017 co-located with 14th ESWC 2017*, 2017.
10. Amna Dridi and Diego Reforgiato Recupero. Leveraging semantics for sentiment polarity detection in social media. *International Journal of Machine Learning and Cybernetics*, Sep 2017.
11. Ronen Feldman. Techniques and applications for sentiment analysis. *Commun. ACM*, 56(4):82–89, April 2013.
12. J Forster and B Entrup. A cognitive computing approach for classification of complaints in the insurance industry. *IOP Conference Series: Materials Science and Engineering*, 261(1):012016, 2017.
13. Thomas Gaillat, Manel Zarrouk, Andre Freitas, and Brian Davis. The ssix corpus: A trilingual gold standard corpus for sentiment analysis in financial microblogs. *11th edition of the Language Resources and Evaluation Conference*, 2018.
14. A. Gangemi, V. Presutti, and D. Reforgiato Recupero. Frame-based detection of opinion holders and topics: A model and a tool. *IEEE Computational Intelligence Magazine*, 9(1):20–30, Feb 2014.
15. J. O. Gutierrez-Garcia and E. López-Neri. Cognitive computing: A brief survey and open research challenges. In *2015 3rd International Conference on Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence*, pages 328–333, 2015.

16. John E. Kelly and Steve Hamm. *Smart Machines: IBM's Watson and the Era of Cognitive Computing*. Columbia University Press, New York, NY, USA, 2013.
17. Bing Liu. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, 2012.
18. Andrés Montoyo, Patricio Martínez-Barco, and Alexandra Balahur. Subjectivity and sentiment analysis: An overview of the current state of the area and envisaged developments. *Decision Support Systems*, 53(4):675 – 679, 2012.
19. Preslav Nakov, Alan Ritter, Sara Rosenthal, Veselin Stoyanov, and Fabrizio Sebastiani. SemEval-2016 task 4: Sentiment analysis in Twitter. In *Proc. of SemEval '16*, San Diego, California, June 2016. ACL.
20. Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008.
21. Diego Reforgiato Recupero, Sergio Consoli, Aldo Gangemi, Andrea Giovanni Nuzzolese, and Daria Spampinato. A semantic web based core engine to efficiently perform sentiment analysis. In Valentina Presutti, Eva Blomqvist, Raphael Troncy, Harald Sack, Ioannis Papadakis, and Anna Tordai, editors, *The Semantic Web: ESWC 2014 Satellite Events*, pages 245–248, Cham, 2014. Springer International Publishing.
22. D.R. Recupero, M. Dragoni, and V. Presutti. Eswc 15 challenge on concept-level sentiment analysis. *Communications in Computer and Information Science*, 548:211–222, 2015.
23. D. Reforgiato Recupero, E. Cambria, and E. Di Rosa. Semantic sentiment analysis challenge at eswc2017. *Communications in Computer and Information Science*, 769:109–123, 2017.
24. Diego Reforgiato Recupero, Valentina Presutti, Sergio Consoli, Aldo Gangemi, and Andrea Giovanni Nuzzolese. Sentilo: Frame-based sentiment analysis. *Cognitive Computation*, 7(2):211–225, Apr 2015.
25. Sara Rosenthal, Noura Farra, and Preslav Nakov. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proc. of the 11th International Workshop on Semantic Evaluation*, SemEval '17, Vancouver, Canada, August 2017. ACL.
26. Hassan Saif, Yulan He, and Harith Alani. Semantic sentiment analysis of twitter. In Philippe Cudré-Mauroux, Jeff Heflin, Evren Sirin, Tania Tudorache, Jérôme Euzenat, Manfred Hauswirth, Josiane Xavier Parreira, Jim Hendler, Guus Schreiber, Abraham Bernstein, and Eva Blomqvist, editors, *The Semantic Web – ISWC 2012*, pages 508–524, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
27. Mikalai Tsytsarau and Themis Palpanas. Survey on mining subjective data on the web. *Data Mining and Knowledge Discovery*, 24(3):478–514, May 2012.
28. Aggeliki Vlachostergiou, George Marandianos, and Stefanos Kollias. From conditional random field (crf) to rhetorical structure theory(rst): Incorporating context information in sentiment analysis. In Eva Blomqvist, Katja Hose, Heiko Paulheim, Agnieszka Ławrynowicz, Fabio Ciravegna, and Olaf Hartig, editors, *The Semantic Web: ESWC 2017 Satellite Events*, pages 283–295, Cham, 2017. Springer International Publishing.

On Finding the Relevant User Reviews for Advancing Conversational Faceted Search

Eleftherios Dimitrakis[1,2], Konstantinos Sgontzos[1,2], Panagiotis Papadakos[1], Yannis Marketakis[1], Alexandros Papangelis[3], Yannis Stylianou[2,3], and Yannis Tzitzikas[1,2]

[1] Institute of Computer Science - FORTH-ICS, Greece
{dimitrakis, sgontzos, papadako, marketak, tzitzik}@ics.forth.gr
[2] Computer Science Department - University of Crete, Greece
[3] Speech Technology Group - Toshiba Research Europe
{alex.papangelis, yannis.stylianou}@crl.toshiba.co.uk

Abstract. Faceted Search (FS) is a widely used exploratory search paradigm which is commonly applied over multidimensional or graph data. However sometimes the structured data are not sufficient for answering a user's query. User comments (or reviews) is a valuable source of information that could be exploited in such cases for aiding the user to explore the information space and to decide what options suits him/her better (either through question answering or query-oriented sentiment analysis). To this end in this paper we introduce and comparatively evaluate methods for locating the more relevant user comments that are related with the user's focus in the context of a conversational faceted search system. Specifically we introduce a dictionary-based method, a word embedding-based method, and one combination of them. The analysis and the experimental results showed that the combined method outperforms the other methods, without significantly affecting the overall response time.

1 Introduction

Faceted Search (FS) is a widely used exploratory search paradigm. It is used whenever the user wants to find the desired item from a list of items (either products, hotels, restaurants, publications, etc). Typically FS offers exploratory search over multidimensional or graph data. However sometimes the structured data are not enough for answering a user's query. User comments (or reviews) is a valuable source of information that could be exploited in such cases for aiding the user to explore the information space and to decide what options suits him/her better. Indeed, user comments/reviews are available in various applications of faceted search, e.g. for hotel booking and in product catalogs.

Enabling the interaction of FS though spoken dialogue, is appropriate for situations where the user cannot (or is not convenient to) use his hands or eyes. In such cases, the user interacts using his voice and provides commands or poses questions. If a question cannot be translated to a query over the structured

resources of the dataset, then the system cannot deliver any answer. In such cases it is reasonable to resort to the available unstructured data, i.e. to users' comments and reviews. Figure 1 illustrates the context. The objective is not to provide the user with a direct answer but first to identify which of the user comments are relevant to the user's question. Direct query answering is reasonable only in cases where, there is a single and credible source of unstructured data (e.g. wikipedia). This is not the case with user comments since they can be numerous, and their content can be conflicting. If we manage to find the relevant comments, then the system could either read these comments to the user, or attempt to apply question answering if the user requests so, or any other kind of analysis, e.g. sentiment analysis as in [2, 14]. In any case spoken dialogue interaction, poses increased requirements on quality, since the system should not "read" irrelevant comments as reading costs user time.

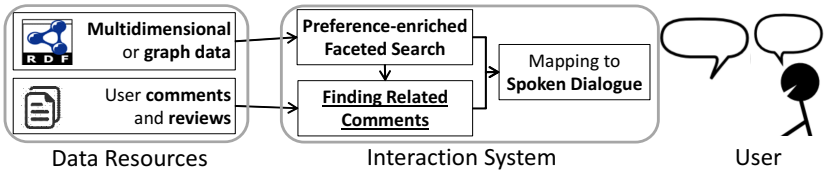

Fig. 1: Finding Related Comments and Conversational Faceted Search

Note that instead of analyzing the user comments for estimating whether a hotel is good or bad as a whole, the interaction that we propose enables the user to get information about the particular aspects or topics that are important for him, e.g. about noise, cleanliness, the quality of the wifi, parking, courtesy and helpfulness of staff, etc. The set of such topics is practically endless and we cannot make the assumption that structured data will exist for all such topics. Therefore, it is beneficial to have systems that are able to exploit associated unstructured data, e.g. user comments and reviews. The problem is challenging because user comments are usually short, meaning that it is hard to achieve an acceptable level of recall. In this paper we focus on this problem, and we introduce methods relying on hand crafted and statistical dictionaries for identifying the relevant comments. In addition we describe an evaluation collection that we have created for comparatively evaluating the introduced methods, as well as an ongoing application and evaluation over a bigger and real dataset. In a nutshell, the key contributions of this paper are: (a) we show how the FS interaction can be extended for exploiting unstructured data in the form of user comments and reviews, and (b) we introduce and comparatively evaluate four methods for identifying the more relevant user comments in datasets related to the task of hotel booking. The rest of this paper is organized as follows: Section 2 presents the required background and related work. Section 3 describes the proposed methods. Section 4 reports experimental results. Finally, Section 5 concludes the paper and discusses directions for future research and work.

2 Background and Related Work

2.1 Background: Faceted Search and PFS

Faceted search is the de-facto standard in e-commerce and tourism services. It is an interaction framework based on a multi-dimensional classification of data objects, allowing users to browse and explore the information space in a guided, yet unconstrained way through a simple visual interface [15]. Features of this framework include: (a) display of current results in multiple categorization schemes (called facets, or dimensions, or just attributes), (b) display of facets and values leading to non-empty results only, (c) display of the count information for each value (i.e. the number of results the user will get by selecting that value), and (d) ability to refine the focus gradually, i.e. it is a session-based interaction paradigm in contrast to the stateless query-and-response dialogue of most search systems. Faceted search is currently the de facto standard in e-commerce (e.g. eBay, booking.com), and its popularity and adoption is increasing. It has been proposed and applied for web searching, for semantically enriching web search results, for patent-search, as well as for exploring RDF and Linked Data (e.g. see [4,16], as well as [19] for a recent survey). The enrichment of faceted search with *preferences*, hereafter *Preference-enriched Faceted Search*, for short PFS, was proposed in [12,20]. PFS offers actions that allow the user to order facets, values, and objects using *best, worst, prefer to* actions (i.e. relative preferences), *around to* actions (over a specific value), or actions that order them lexicographically, or based on their values or count values. Furthermore, the user is able to *compose* object related preference actions, using *Priority, Pareto, Pareto Optimal* (i.e. skyline) and other. The distinctive features of PFS is that it allows expressing preferences over attributes, whose values can be hierarchically organized (and/or multi-valued), it support preference inheritance, and it offers scope-based rules for resolving automatically the conflicts that may arise. As a result the user is able to restrict his current focus by using the faceted interaction scheme (hard restrictions) that lead to non-empty results, and rank the objects of his focus according to the expressed preferences. Recently, PFS has been used in various domains, e.g. for offering a flexible process for the identification of fish species [17], as a Voting Advice Application [18], as well as, for data that contain also geographical information [6].

2.2 Related Works

Conversational Faceted Search Only a few works exist that involve speech interfaces on top of the faceted search paradigm: [3] exploits a speech user interface over facets that index audio metadata associated with audio content (that system is used for the Spoken Web, an alternative to WWW based on audio content, and the associated Mediaeval Spoken Web Search Task), while a faceted browser over Linked Data is described in [7], where commands in natural language are translated to SPARQL queries. To the best of our knowledge though, the only work that combines spoken dialogue systems with faceted search is the

one presented in [13], where the described LD-SDS system is limited to spoken dialogues over structured datasets (expressed in RDF). In this work we extend conversational faceted search for exploiting also available unstructured data (e.g. user reviews). Note that, tackling the same problem using only a single large-scale source of unstructured data, e.g. Wikipedia (as described in [1]), is much easier since in that case we do not have the source selection problem (selection of user comments in our case), and the source contains many and long texts, therefore it is not difficult to achieve a good recall level.

Similar Tasks Two similar tasks, as regards the text size, from the area of Question Answering are: (1) *Machine Comprehension (MC)* which aims at identifying the answer boundaries from a given text passage and an input question (e.g. [1] performs MC over Wikipedia), and (2) *Answer Sentence Selection* which aims at identifying the right sentence from a list of candidate sentences, given an input question (e.g. as in [21]).

3 The Proposed Approach

In §3.1 we describe an extension of the interaction of PFS for exploiting also associated unstructured data, and in §3.2 we focus on the problem of finding the relevant comments.

3.1 The Interaction

The user interacts with the system using actions corresponding to PFS actions, i.e. actions that correspond either to hard constraints (i.e. filters), or soft constraints (i.e. preferences). We shall use the term *ofocus* to refer to the restricted set of objects (those after applying all filters), and *pfocus* to refer to the first bucket of the focus, that contains the more preferred objects. If the cardinality of either of the above sets is below a configurable threshold θ (say 10), then if the user's questions cannot be answered by the structured dataset, the system resorts to the user comments for this. Note that if at some point in the interaction, the user's focus is big (i.e. $\min(|ofocus|, |ofocus|) > \theta$) and the user asks a question that cannot be answered by the structured dataset, then the system suggests the user to "first refine the focus" in the sense that it is not useful to ask questions of the form "quiet hotel in Rome", or "hotels with fast wifi in London". In other words, we could say that the system enters this mode in the so-called "End Game" phase of faceted search [15]. This choice has several benefits:
(a) *Applicability*: It can applied without requiring the comments to be indexed a priori, and this enables the application of this model over RSS feeds and blog comment hosting services (e.g. Disqus).
(b) *Efficiency:* Since the analysis will be done only for the comments of the hotels in the focus, it is feasible to make this analysis at real time.
(c) *Less Noise, Better Quality:* For the same reason, as in (b), the quality of the retrieved comments is expected to be higher in comparison to the quality of retrieval over the entire set of comments (of all hotels).

3.2 Finding the Relevant Comments

We shall use a scoring function for estimating the relevance between an input question q and each user review r_i, where $1 \leq i \leq \theta$. Below we introduce four scoring methods: (I) a Baseline, (II) a WordNet-based, (III) a Word2vec-based, and (IV) a combination of (II) and (III).

WordNet [11] is lexical database for the English language comprising 166,000 (f, s) pairs, where f is a word-form and s the set of words that have the same sense, that also includes relations between words and senses (like Synonymy, Antonymy, Hypernymy etc.). *Word2Vec* [10] is a method for transforming individual words into vectors of low dimensionality (it is low in comparison a $|words|$-dimensionality), e.g. 300, so that their distances reveal their semantic association (these representations are derived by training a two-layer neural network). The motivation for the selection of the above methods is their ability to capture semantically relevant reviews beyond the trivial task of exact string matching, and their rich and domain-agnostic vocabulary.

The process for identifying the more relevant comments, in any of the four I-IV methods, consists of the following steps:
1) For each review r_i we split its text into individual sentences and get a set of sentences r_{ij}, where $1 \leq j \leq s$ and s is the number of sentences in each review. In this way, we can score the reviews based on the maximal scored sentence.
2) Apply tokenization, removal of stop-words and punctuations, as well as lemmatization (using Stanford CoreNLP [8]) both to the input question q and each associated sentence r_{ij} of r_i. Let denote the result by q_words and r_{ij}_words respectively.
3) Construct the method-related representation of q and each r_{ij} (it will be described below).
4) Score and rank each review based on the defined relevancy formula.

Below we describe the representation and the scoring formula for each method.

I) **Baseline:** Here we just compute the maximum Jaccard Similarity between the q_words and the corresponding r_{ij}_words sets:
$S(q, r_i) = \max_{\forall r_{ij} \in r_i} JaccardSim(q\_words, r_{ij}\_words)$

II) **WordNet:** In this method we construct WordNet-based representations for the q and r_{ij} sets. Specifically, for each word in q_words and r_{ij}_words we take the union of the synonyms, antonyms and hypernyms, denoted by $wordNet(q)$ and $wordNet(r_{ij})$ respectively, as extracted from the WordNet. The final score is defined again using the maximum Jaccard Similarity as:
$S(q, r_i) = \max_{\forall r_{ij} \in r_i} WNS(q, r_{ij})$,
where $WNS(q, r_{ij}) = JaccardSim(wordNet(q), wordNet(r_{ij}))$.

III) **Word2vec:** This method exploits the word2vec embeddings available in the GoogleNews 300-dimensional pre-trained model[4]. Specifically, we get the

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

word2vec vector representations of all words in q_words and r_{ij}_words, denoted by $word2vec(q)$ and $word2vec(r_{ij})$ respectively. Then we apply the Word Movers Distance (WMD) [5] which calculates the minimum distance (in the vector space) between the embedded words of the two sets. The score is defined as:
$S(q, r_i) = \max_{\forall r_{ij} \in r_i} WMS(q, r_{ij})$,
where $WMS(q, r_{ij}) = 1 - WMD_n(word2vec(q), word2vec(r_{ij}))$ and WMD_n denotes the normalized distance calculated by the division with the max WMD over all comments.

IV) **WordNet and Word2vec:** Here we combine the two previous methods through a weighted sum, reaching to the following definition of score:

$$S(q, r_i) = w_{wN} * \max_{\forall r_{ij} \in r_i} WNS(q, r_{ij}) + w_{w2v} * \max_{\forall r_{ij} \in r_i} WMS(q, r_{ij})$$

where $w_{wN}, w_{w2v} \in [0, 1]$ and $w_{wN} + w_{w2v} = 1$.

4 Evaluation

4.1 Evaluation over the Collection FRUCE

We constructed a small evaluation collection in order to compare the presented methods. The collection consists of 40 hand crafted user reviews/comments related to hotels $(c_1, ..., c_{40})$ and 2 manually crafted queries (q_1 and q_2) related to the topic of *noise*. The complete list of comments is web accessible[5] and the queries are the following: q_1 = "Has anyone reported a problem about noise?", q_2 = "Is this hotel quiet?".

For the needs of the evaluation we manually judged the relevance of the collection's reviews to each query. Specifically, each review c_i is labeled with 1 if it is relevant, and with 0 otherwise. The relevant/irrelevant ratio in the collection is 1/3.

Quality. We measured the *mean* $R - Precision$ and *mean* $AveP$ over the two queries q_1 and q_2 for all methods. Specifically, for the IV method we computed various weights combinations and chose the model that achieved the highest *mean* $AveP$. Note that methods II and III correspond to the pairs $(w_{wN} = 1.0, w_{w2v} = 0.0)$ and $(w_{wN} = 0.0, w_{w2v} = 1.0)$ respectively. In our case the maximizing weights were found to be $w_{wN} = 0.7$ and $w_{w2v} = 0.3$ with *mean* $AveP = 0.569$ and *mean* $R-Precision = 0.649$. The corresponding scores for method II were *mean* $AveP = 0.398$ and *mean* $R-Precision = 0.449$, while method III achieved *mean* $AveP = 0.366$ and *mean* $R - Precision = 0.4$ (the precision of Word2vec-based methods in analogous challenges [9] is around 55%, i.e. similar to what we measured in our setting). Finally, IV outperforms both II, III, while II slightly outpoints III. As expected, all of the above models outperformed our baseline (*mean* $AveP = 0.05$ and *mean* $R - Precision = 0.05$) as shown in Table 1. We have to stress though that the results of methods II

[5] at `http://www.ics.forth.gr/isl/sar/resources/dataset/fruce`

and IV could be further improved by combining other thesaurus with WordNet or an updated version of WordNet, since WordNet currently fails to provide the synonyms, hypernyms and antonyms of many words. Further, since we currently consider all possible senses of a word in the WordNet based approach, we might be introducing wrong terms in the $wordNet(q)$ and $wordNet(r_{ij})$ set. This problem can possible be avoided with proper sense identification methods.

Method	Mean AveP	Mean R-Precision
I	0.05	0.05
II	0.398	0.449
III	0.366	0.4
IV	**0.569**	**0.649**

Table 1: Mean Average Precision of methods I-IV.

Method	Total time (ms)	Aver. time (ms)
I	141	3
II	797	19
III	**47**	**1**
IV	546	13

Table 2: Time for computing the score of 40 reviews for each method.

In addition, we plot a 2D diagram for each of the three models II, III, IV (baseline excluded), where the *y-axis* represents the computed score for (r_i, q_i) and the *x-axis* indicates its true binary relevance. The plots are shown in Figure 2. We can observe that the points are not separable by a threshold in any of the figures (parallel line to *x-axis*). However, it is obvious that the IV approach clearly improves the separation, preserving higher scores to the true relevant reviews, like III, and lower scores to the non-relevant ones, like II.

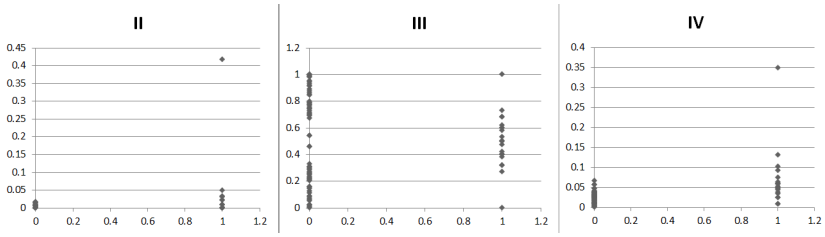

Fig. 2: Distribution of query-review pairs as a function of their calculated (floating point) and true binary relevance score for methods II, III, IV.

Efficiency. All experiments were performed using a 16GB RAM machine. Regarding speed efficiency, it is worth measuring: a) the required time for loading the appropriate resources (Dataset, WordNet, Word2Vec) (i.e. *Init Time*), and b) the required time for computing the similarity score of one query-review pair (i.e. *Execution Time*). Note that the *Init Time* cost has to be paid only once, while *Execution Time* affects the user interaction.

Regarding *Init Time*, the most time consuming resource is Word2Vec due to its enormous size (491,061 ms), followed by the loading of the FRUCE dataset (39,149 ms). The WordNet dictionary loads almost instantly ($63ms$). The *Execution Time* on the other hand is very fast for all methods (only 13 ms on average). We only need about 1.5 seconds for analyzing and scoring 100 reviews

with 15 words on average. Table 2 shows the *Execution Time* for computing the scores of the 40 reviews for all methods (the minimum values are in bold).

4.2 Experiments over a Real Dataset

We also evaluated the proposed methods over a real dataset, that we scrapped from a travel website. This specific dataset contains information about 382 different hotels located in 4 different cities (Kyoto, Tokyo, Osaka, Kobe) of Japan. The extracted data are logically structured in facets so that they can be directly plugged into the system, containing the following types of information: (a) boolean values, used for describing the facilities of a hotel (e.g. free of charge wifi, free parking, etc.), (b) numerical values (integers or floats) for describing quantitative values (e.g. price, review rating, distance from various points of interest, etc.), (c) geographic values for describing the location and (d) textual values. In the last category there are also comments that review hotels, which are categorized into comments with a positive and negative aspect. We would like to remark that almost all (more than 23 thousand) review comments that we have extracted contain both a positive and a negative part. Table 3 shows the total number of hotels and the average number of comments per hotel for the 4 different cities of Japan.

City	hotels	avg num. of comments per hotel
Kyoto	100	71
Osaka	100	65
Tokyo	100	71
Kobe	82	33
Total	382	61

Table 3: The Japan hotels dataset containing more than 23,000 comments

Efficiency. The time required to load the user reviews is 186,769 ms. For evaluating the execution time, we measured the required time for analysing and scoring (according to the q_1 and q_2) 2,000 randomly selected reviews, returning the 10 most highly ranked ones. The minimum, maximum and average times were 21 ms, 6,427 ms and 56 ms respectively (on average each review has 48 words), and the total time was 113,870 ms. It follows that the proposed method is acceptable in terms of efficiency. Specifically, if we assume that we have 3 hotels in the current user focus and the average number of reviews per hotel is 61 (as shown in Table 3), we can score all reviews in around 10 secs.

Quality. Since the reviews are not annotated with binary relevance scores for the two used queries, it is difficult to evaluate the quality of the scoring methods on this collection. Annotating the whole collection is a laborious and time consuming task. However we have started to manually annotate a part of the full reviews for the two queries that we have used in the FRUCE Collection. For the time being, we have marked 71 distinct comments, and identified 66 relevant and 76 irrelevant (c_i, q_i) pairs. The average top-2 precision of the IV method for the 2 queries by considering *only* the subcollection of 71 human judged comments is 0.5, while the average R-precision ($R = 33$) is again 0.5. We have noticed that

we would get higher results if the comments were clean, in the sense that the collection has several spam comments that affect negatively the results. Currently, we are in the process of cleaning the collection.

5 Conclusion

In the context of Faceted Search quite often the structured data are not enough for answering a users query. In such cases the system could resort to related textual comments (posed in natural language) for identifying those that could be exploited for helping the user. This requires finding the most relevant comments that (a) are associated with the most preferred objects, and (b) are related to a user's question. Moreover, spoken dialogue interaction poses increased requirements on quality, in order to avoid wasting user's time by reading irrelevant comments. To this end, we introduced a dictionary-based method that uses WordNet, a word embedding-based method, specifically Word2vec, and one that combines both. The analysis and the experimental results showed that the key result is that without dictionaries (either human-made or statistical ones), the effectiveness of retrieving the relevant comments is very low even in a small dataset. Specifically, the baseline method achieved $mean\ AveP = 0.05$ and $mean\ R-precision = 0.05$. However the method that uses both WordNet and Word2vec outperforms every other method with $mean\ AveP = 0.569$ and $mean\ R-precision = 0.649$, taking on average only 13 ms to score a review. We believe that the proposed method can be applied in several domains and for various tasks, from booking services to product selection. As part of our future work we plan to: (a) continue the quality evaluation over the real dataset, (b) extend the system described in [13] with this functionality, (c) investigate the applicability of comparative opinion mining and query-oriented sentiment analysis, and (d) investigate how we could exploit external sources in cases where even the user comments/reviews are not sufficient for answering a user's question.

References

1. D. Chen, A. Fisch, J. Weston, and A. Bordes. Reading wikipedia to answer open-domain questions. *CoRR*, abs/1704.00051, 2017.
2. K. Cortis, A. Freitas, T. Daudert, M. Hürlimann, M. Zarrouk, S. Handschuh, and B. Davis. Semeval-2017 task 5: Fine-grained sentiment analysis on financial microblogs and news. In *Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval@ACL 2017, Vancouver, Canada, August 3-4, 2017*, pages 519–535, 2017.
3. M. Diao, S. Mukherjea, N. Rajput, and K. Srivastava. Faceted search and browsing of audio content on spoken web. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1029–1038. ACM, 2010.
4. S. Ferré. Sparklis: an expressive query builder for sparql endpoints with guidance in natural language. *Semantic Web*, 8(3):405–418, 2017.
5. M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger. From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966, 2015.

6. P. Lionakis and Y. Tzitzikas. Pfsgeo: Preference-enriched faceted search for geographical data. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 125–143. Springer, 2017.
7. B. L. López-Ochoa, J. L. Sánchez-Cervantes, G. Alor-Hernández, M. A. Abud-Figueroa, B. A. Olivares-Zepahua, and L. Rodríguez-Mazahua. An architecture based in voice command recognition for faceted search in linked open datasets. In *International Conference on Software Process Improvement*, pages 174–185. Springer, 2017.
8. C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60, 2014.
9. T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
10. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.
11. G. A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, Nov. 1995.
12. P. Papadakos and Y. Tzitzikas. Comparing the effectiveness of intentional preferences versus preferences over specific choices: a user study. *International Journal of Information and Decision Sciences*, 8(4):378–403, 2016.
13. A. Papangelis, P. Papadakos, M. Kotti, Y. Stylianou, Y. Tzitzikas, and D. Plexousakis. Ld-sds: Towards an expressive spoken dialogue system based on linked-data. In *Search Oriented Conversational AI, SCAI 17 Workshop (co-located with ICTIR 17)*, 2017.
14. G. Petrucci and M. Dragoni. An information retrieval-based system for multi-domain sentiment analysis. In F. Gandon, E. Cabrio, M. Stankovic, and A. Zimmermann, editors, *Semantic Web Evaluation Challenges*, pages 234–243, Cham, 2015. Springer International Publishing.
15. G. M. Sacco and Y. Tzitzikas. *Dynamic taxonomies and faceted search: theory, practice, and experience*, volume 25. Springer Science & Business Media, 2009.
16. E. Sherkhonov, B. C. Grau, E. Kharlamov, and E. V. Kostylev. Semantic faceted search with aggregation and recursion. In *International Semantic Web Conference*, pages 594–610. Springer, 2017.
17. Y. Tzitzikas, N. Bailly, P. Papadakos, N. Minadakis, and G. Nikitakis. Using preference-enriched faceted search for species identification. *International Journal of Metadata, Semantics and Ontologies*, 11(3):165–179, 2016.
18. Y. Tzitzikas and E. Dimitrakis. Preference-enriched faceted search for voting aid applications. *IEEE Transactions on Emerging Topics in Computing*, PP(99):1–1, 2016.
19. Y. Tzitzikas, N. Manolis, and P. Papadakos. Faceted exploration of rdf/s datasets: a survey. *Journal of Intelligent Information Systems*, pages 1–36, 2016.
20. Y. Tzitzikas and P. Papadakos. Interactive exploration of multidimensional and hierarchical information spaces with real-time preference elicitation. *Fundamenta Informaticae*, 20:1–42, 2012.
21. L. Yu, K. M. Hermann, P. Blunsom, and S. Pulman. Deep learning for answer sentence selection. *CoRR*, abs/1412.1632, 2014.

What does it mean to be a Wutbürger? A first exploration.

Manfred Klenner

Institute of Computational Linguistics
Andreasstrasse 15, 8050 Zurich, Switzerland
klenner@cl.uzh.ch

Abstract. In this paper, we undertake an attempt to characterize the world view of what are called Wutbürger in Germany, that is citizen who are enraged by the current political and social situation. In order to find out what makes a Wutbürger a Wutbürger, we analyze Facebook posts on the basis of a lexical resource where nouns, adjectives and verbs are classified according to Plutchik's primary emotions. We also introduce new polar roles of verbs that help to identify the writer perspective. This way, we are able to identify targets and the Wutbürger's stance towards them. As textual data, we utilize about 100,000 Facebook posts of a German right-wing party whose members are obvious exemplars of the notion of a Wutbürger.

1 Introduction

In a number of western societies populism has (re)entered the scene and especially rampages in the social media. Hate speech, shit storms etc. are extreme forms of such undemocratic tendencies. In Germany, the notion of a *Wutbürger* has been coined, that is, citizen who are disappointed by the government and the social situation and their (verbal) behavior seem to be driven by rage (German *Wut*). A new, right-wing party evolved, the AfD (Alternative für Deutschland). We have access to about 100,000 Facebook posts of the AfD including reader comments that mostly stem from AfD proponents - who clearly form a subset of German Wutbürger. Our research question was: Can we find out, how a Wutbürger perceives the world and what, after all, is the objective of his Wut.

A first step towards this goal is to measure the emotional fingerprint of the texts produced by Wutbürger and compare it to the fingerprint of a related text genre. We use the Tübinger (German newspaper) Treebank (TüBa-D/Z) [11] as a reference corpus. In order to compare the emotional load of the AfD texts (303,563 sentences) and the TüBa-D/Z texts (95,595 sentences), we perform a lexicon-based analysis. That is, we count the primary emotions by the use of words that are indicative of these emotions. This is a straightforward approach, but it should tell us reliably what the prevalent emotions in these texts are and whether the AfD texts are more loaded than the newspaper reference texts.

The emotional fingerprint does not tell us anything about the world view of a Wutbürger: who are his proponents and who are his opponents? We would like

to exploit the idea that the identification of these targets can be supported and accomplished by a more fine-grained classification of lexical items. We not only assign primary emotions to words (verbs, nouns and adjectives), we also identify those words that have an implicit writer perspective and explicate which one it is. For instance, the adjective *ineffable* in a phrase like *the ineffable chancellor* expresses that the writer has a negative attitude towards the referent of the noun and a sentence like *Merkel jerks the German citizen around* allows to infer that the writer believes that the referent at subject position is an immoral actor, a cheater one might say. Also, the direct object is perceived as a victim of the cheater. Since the writer is against the cheater, he is in favor of the victim.

2 Lexical Resources

Starting with the freely available lexicons described in [2][1] and [4][2] we identified those verbs, nouns and adjectives that have an emotional dimension (e.g. *to love, to hate, gratitude, joy, pleasant, happy*). We then classified each of the 168 verbs, 225 nouns and 300 adjectives according to Plutchik's [7] eight primary emotions which are *anger, fear, sadness, disgust, surprise, anticipation, trust*, and *joy*. This was done by two annotators, who achieved a Kappa value of κ 0.73. Furthermore, we annotated those verbs, nouns and adjectives that refer to moral, e.g. *to lie, donation*. See Figure 1 for an overview (*pos, neg* are shortcuts for *positive, negative* respectively). Kappa was $\kappa = 0.66$.

	pos emotion	neg emotion	pos moral	neg moral	pos factual	neg factual	#
verb	49	119	15	71	553	1170	1977
adj	118	182	286	569	1010	1103	3268
noun	91	134	104	436	663	1229	2657

Fig. 1. Lexicon Overview: Word Frequencies

The columns for *factual* denote words that are positive or negative without reference to either (a particular) emotion or moral. We could say that they are positive or negative on a factual level. For instance *to sicken, recover, congratulate* are examples of such verbs, whereas *mistake, disease, transparency, security, right, wrong* are examples for such nouns and adjectives. This is a crucial distinction: such words do not indicate a writer perspective, but the contribute to polarity decisions, nevertheless.

We took the 254 verbs classified as either belonging to the emotion or moral dimension as a basis for further annotations. We identified 58 verbs with a very strong writer perspective either on the actor or the experiencer role or on both (*to cheat, to jerk sb around*). We then coined for the six verb classes derived that way special role labels. The set of agent roles is: prole, baiter, hater, torturer, hypocrite, choleric. Experiencer roles are sufferer and victim. To give an example: the verb *flare up* (aufbrausen) bears the emotion *anger* and the semantic

[1] http://bics.sentimental.li/files/8614/2462/8150/german.lex
[2] https://pub.cl.uzh.ch/projects/opinion/lrec_data.txt

role of the subject is that of *choleric*. Our hypothesis is that these roles better capture the writer perspective, since they express how the writer conceptualizes these referents. Note that we assign these roles to subcategorization frames, not to verbs. We specified these verbs along the line proposed by [4]. That is, we modeled the various subcategorization frames of a verb and assigned it a polar effect (positive or negative) and for some of the verbs also a dedicated polar role (sufferer, torturer etc.). We thus were able to find out who the AfD believes to be a torturer, a baiter etc. and who suffers from the situation described.

3 Corpus Statistics and Lexical Coverage

Our present endeavor is basically one that exploits an existing, but carefully refined and augmented lexical resource. Especially our new verb classes with a new kind of polar roles are meant to make the writer perspective more nuanced. We are at the very beginning of a sophisticated study. At the moment, however, there is no gold standard and thus no machine learning involved.

We found 9,012 verb types in the Facebook posts, which gave us altogether 419,034 verb tokens in 303,563 sentences (word tokens altogether: 5,249,613). The 1052 verb types of our lexicon found in the data (61 verbs did not occur), amounts to 83,658 verb tokens, which is - taking into account that one sentence might have more than one model verb - about 25% coverage (a model verb in each 4th sentence). Our verb resource seems to have a good coverage, thus. If we just look at moral verbs, we get 11,153 hits, 64 of the 85 moral verb types do occur in the posts. The 168 emotional verbs occur with a frequency of 17,102.

In order to quantify the emotional load of the Facebook posts, we used the Tübinger Treebank (TüBa-D/Z) as a reference corpus. The TüBa-D/Z comprises 95,595 sentences. The coverage of our verb resource is again quite good: we found 930 verb types with 22,679 verb tokens, which is a coverage of 23.79% (again almost each 4th sentence bears a model verb).

4 Emotional Fingerprint

We use our emotion lexicon in order to diagnose the emotional load of the AfD posts. In Fig. 2, we compare the AfD posts and the newspaper text wrt. to the emotions present. We determined the frequency of words belonging to a particular emotion and normalized by the total number of emotion words (found in the posts) (left hand side), and by the number of sentences (right hand side), respectively.

As we can see from Fig. 2, fear is the most prominent emotion of a Wutbürger and not anger (a prestage of *rage*) while at the same time *sadness* is not a prevalent emotion of a Wutbürger. All other emotions are almost identically distributed in both, AfD posts and newspaper text. Our expectation, namely that the AfD posts would have a higher emotional load than news texts, was not confirmed.

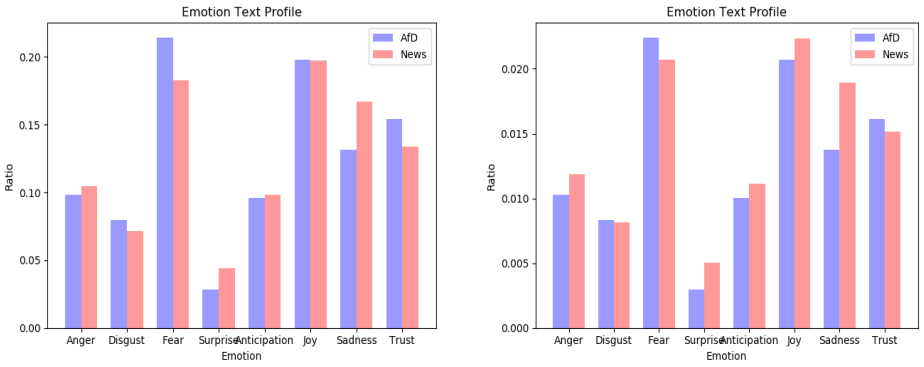

Fig. 2. Emotional Fingerprints of AfD Facebook Post and a Newspaper Corpus

We also had a look at the moral dimension. The TüBa-D/Z refers to 7490 nouns and adjectives classified as positive (32%) or negative (60%) from a moral perspective, i.e. 7.8% of the sentences refer to that dimension. In the AfD texts, 32167 tokens were found, which is about 10.6% (73% negative, 27% positive). This clearly shows that (negative) moral argumentation is a central attitude of a Wutbürger. If we have a look at verbs the picture is similar: 1.6% of the news texts contain a moralizing verb, whereas 2.3% of the AfD posts do so.

5 Target Identification and Stance Analysis

A *polar role* is the label for the logical subject (agent) or object (theme, patient, experiencer) that indicates the positive or negative role its filler plays. The inventory of polar roles is not fixed, yet. We have defined a couple of fine-grained polar roles that are meant to indicate a more nuanced writer perspective. These roles are *baiter, hater, choleric, hypocrite, prole, torturer and sufferer, victim.* The definition of these roles is straightforward: we just had to fix the corresponding verbs and determine which semantic role bears which polar role. Take the polar role *prole*. There is a number of verbs in German (we have identified 18) that indicate that the writer implicitly classifies the agent of such a verb as a prole (*anlabern (to chat so up), anpöbeln (to accost sb)*). Thus, the agents of such verbs are negative targets from the point of view of the writer. It turns out that in the AfD texts journalists, do-gooder, politicians, asylum seekers, the print media are, among others, conceptualized as proles.

In order to derive these writer perspectives, we have parsed the AfD posts with a dependency parser [10], normalized the parse trees (e.g. passive voice) and extracted the filler of the polar roles. This way we found e.g. that (the targets) Merkel, the German government, the police and the press are baiter, hater and torturers etc. The AfD, the German citizen and women were victims and sufferer. Clearly, these lists are not perfect. There are (third person) pronouns in it and also words denoting non-actors. The goal of this explorative study was to get a

proof of concept not a full-fledged evaluation. Nevertheless, we have carried out a small evaluation in order to find out where the noise comes from. We randomly took 50 sentences where the German chancellor Angela Merkel was the logical subject of a morally negative verb (e.g. *cheat, threaten, diss, violate*) and 50 where the AfD was the experiencer or patient of such a verb. Only 9 out of the 100 decisions were wrong due to 4 parsing errors and 5 modal constructions that erroneously passed our modal filter.

An interesting finding concerns the role of the AfD (i.e. Wutbürger) itself. If we look at those who are hated, we get: Arabs, strangers, Merkel, Muslims, comrads but also Germany and the AfD. A closer inspections reveals that the Wutbürger do not disguise or veil their rage. They use verbs with AfD (or *I* or *we*) as agents that indicate that they are haters.

Our verb resource also allows for more sophisticated inferences. We have coined the notion of a violator of morality for the following set of actors: the set of actors of a verb that casts a negative effect on its object which is - according to the polarity lexicon - positive:

$\lambda X.(\exists$ *Verb, Y: subj(Verb,X)*$\wedge$*effect(Verb,obj,neg)*$\wedge obj(Verb, Y) \wedge polarity(Y, pos))$

An example is *Merkel destroys the security of Germany* where *security* is positive and *destroy* casts a negative effect on the direct object (obj)

If we, however, change *polarity(Y,pos)* to *polarity(Y,neg)* than we get a strong proponent of the AfD: to disapprove something negative is positive.

6 Related Work

One topic of this paper is lexicon-based, document-level emotion detection. For an overview of similar approaches see e.g. [1]. We have specified the first German emotion lexicon, where words are associated with primary emotions and - if applicable - writer perspectives.

The role verbs play in sentiment analysis and stance detection has received increased attention over the last years, cf. [6], [9], [3], [8], [5]. The main difference to our German verb resource is that we not only specify the polar effects (positive or negative) a verb casts on its semantic roles, we also strive to assign fine-grained role labels such as torturer etc. Again this is meant to allow for a finer nuanced writer perspective, which not only helps to identify targets, but also the stance taken towards those targets. Another distinctive feature is that we combine bottom-up and top-down information in order to derive stance (see last section).

7 Conclusions

We have introduced two new resources for German: a fine-grained verb resource with polar roles that reveal the writer perspective, and an emotion lexicon where words are classified as one of eight primary emotions. Also words related to moral are specified. We used this in order to find out whether Wutbürger texts do have a

clear emotional fingerprint compared to news texts. We found that *fear* (and not *Wut*, i.e. *rage*) is the prevalent emotion and that Wutbürger significantly more often argue on the basis of moral than a reference newspaper corpus. Another insight is that Wutbürger do not hide their rage (in their own sentences they often occupy negative polar roles such as hater). More sophisticated search pattern on the basis of top down and bottom up restrictions give rise to interesting inferences (someone who disapproves something positive is a violator of morality). All this is meant as a first explorative study: is our lexicon large enough to be useful, is our fine-grained verb resource broadly applicable. A fuller answer to the question raised in the title must await a thorough empirical investigation.

References

1. Haji Binali, Chen Wu, and Vidyasagar Potdar. Computational approaches for emotion detection from text. In *Proceedings of IEEE Intern. Conf. on Digital Ecosystems and Technologies*, 2010.
2. Simon Clematide and Manfred Klenner. Evaluation and extension of a polarity lexicon for German. In *Proceedings of the First Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA)*, pages 7–13, 2010.
3. Lingjia Deng and Janyce Wiebe. Joint prediction for entity/event-level sentiment analysis using probabilistic soft logic models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 179–189, 2015.
4. Manfred Klenner and Michael Amsler. Sentiframes: A resource for verb-centered German sentiment inference. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, may 2016. European Language Resources Association (ELRA).
5. Manfred Klenner, Don Tuggener, and Simon Clematide. Stance detection in Facebook posts of a German right-wing party. In *LSDSem 2017/LSD-Sem Linking Models of Lexical, Sentential and Discourse-level Semantics*, April 2017.
6. Alena Neviarouskaya, Helmut Prendinger, and Mitsuru Ishizuka. Semantically distinct verb classes involved in sentiment analysis. In *IADIS AC (1)*, pages 27–35, 2009.
7. Robert Plutchik. *A general psychoevolutionary theory of emotion*. Academic press, NewYork, 1980.
8. Hannah Rashkin, Sameer Singh, and Yejin Choi. Connotation frames: A data-driven investigation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, Berlin, Germany, Angust 2016.
9. Kevin Reschke and Pranav Anand. Extracting contextual evaluativity. In *Proc. of the Ninth International Conf. on Computational Semantics*, pages 370–374, 2011.
10. Rico Sennrich, Martin Volk, and Gerold Schneider. Exploiting synergies between open resources for German dependency parsing, POS-tagging, and morphological analysis. In *Recent Advances in Natural Language Processing (RANLP 2013)*, pages 601–609, September 2013.
11. Heike Telljohann, Erhard W. Hinrichs, Sandra Kübler, Heike Zinsmeister, and Kathrin Beck. Stylebook for the Tübingen treebank of written German. Technical report, Universität Tübingen, Seminar für Sprachwissenschaft, 2009.

A Dataset for Detecting Irony in Hindi-English Code-Mixed Social Media Text

Deepanshu Vijay*, Aditya Bohra*, Vinay Singh, Syed S. Akthar, and Manish Shrivastava

Language Technology Research Centre, International Institute of Information Technology, Hyderabad,
{deepanshu.vijay,aditya.bohra,vinay.singh, syed.akhtar}@research.iiit.ac.in
m.shrivastava@iiit.ac.in

Abstract. Irony is one of many forms of figurative languages. Irony detection is crucial for Natural Language Processing (NLP) tasks like sentiment analysis and opinion mining. From cognitive point of view, it is a challenge to study how human use irony as a communication tool. While relevant research has been done independently on code-mixed social media texts and irony detection, our work is the first attempt in detecting irony in Hindi-English code-mixed social media text. In this paper, we study the problem of automatic irony detection as a classification problem and present a Hindi-English code-mixed dataset consisting of tweets posted online on Twitter. The tweets are annotated with the language at word level and the class they belong to (Ironic or Non-Ironic). We also propose a supervised classification system for detecting irony in the text using various character level, word level, and structural features.

Keywords: code-mixing, language detection, linguistics, svm, random forest, hate-speech.

1 Introduction

Irony is a subtle form of humor, where there is a gap between the intended meaning and the literal meaning. Even though it is a widely studied linguistic phenomenon, no clear definition seems to exist [5]. Irony detection is a difficult task as irony often has ambiguous interpretations. Apart from it's importance in sentiment analysis and opinion mining, irony detection is also vital in the areas of medical care and security [6]. Previous research related to this task has mainly been focused on monolingual texts [18, 2, 8, 5] due to the availability of large-scale monolingual resources. Popularity of opinion-rich online resources like review forums and microblogging sites has encouraged users to express and convey their thoughts all across the world in real time. In multilingual societies like India, users often interchange between two or more languages while communicating online.

* These authors contributed equally to this work.

Code-Mixing (CM) is a natural phenomenon of embedding linguistic units such as phrases, words or morphemes of one language into an utterance of another [13–15, 4]. English and Hindi are two of the most widely used languages in the world and to the best of our knowledge currently there are no online Hindi-English code-mixed resources available for detecting irony.

Following are some instances of Hindi-English code-mixed tweets. It can be observed that **T1** and **T2** contain irony while **T3** is a non-ironic tweet.

T1 : *"Wo ek teacher hai tab bhi life ke test mein fail ho gaya! Hahaha such irony :D"*
Translation : "He is a teacher yet he failed in the test of life! Hahaha such irony :D."

T2 : *"The kahawat 'old is gold' purani hogaee. Aaj kal ki nasal kehti hai 'gold is old', but the old kahawat only makes sense. #MindF #Irony."*
Translation : "The saying 'old is gold' is old. Today's generation thinks 'gold is old' but only the old one makes sense. #MindF #Irony. "

T3 : *"mere single hone ke bawzood mujhe ye nahi pata tha aaj rose day he #irony."*
Translation : "Inspite of me being single, I didn't know today is rose day #irony."

The structure of the paper is as follows. In Section 2, we review related research in the area of code mixing and irony detection. In Section 3, we describe the corpus creation and annotation scheme. In Section 4, we present our system architecture which includes the pre-processing steps and classification features. In Section 5, we present the results of experiments conducted using various character-level, word-level and structural features. In the last Section, we conclude our paper, followed by future work and references.

2 Background and Related Work

[11] performed analysis of data from Facebook posts generated by English-Hindi bilingual users. Analysis depicted that significant amount of code-mixing was present in the posts. [21] formalized the problem, created a POS tag annotated Hindi-English code-mixed corpus and reported the challenges and problems in the Hindi-English code-mixed text. They also performed experiments on language identification, transliteration, normalization and POS tagging of the dataset. [3] addressed the problem of shallow parsing of Hindi-English code-mixed social media text and developed a system for Hindi-English code-mixed text that can identify the language of the words, normalize them to their standard forms, assign them their POS tag and segment into chunks. [19] addressed

the problem of language identification on Bengali-Hindi-English Facebook comments. They annotated a corpus and achieved an accuracy of 95.76% using statistical models with monolingual dictionaries. [12] developed a Question Classification system for Hindi-English code-mixed language using word level resources such as language identification, transliteration, and lexical translation. [1, 16] performed Sentiment Identification in code-mixed social media text.
[18] proposed an algorithm for separating ironic from non-ironic similes in English, detecting common terms used in this ironic comparison. [8] presented a corpus of Italian tweets which consisted of 25.450 tweets among which 12.5% tweets were ironic and 87.5% tweets were non-ironic. They evaluated their dataset using two systems. The first system relies on lexical and semantic features characterising each word of a Tweet. The second system exploits words occurrences (BOW approach) as features useful to train a Decision Tree. [2] proposed a model to detect irony in English Tweets, pointing out that skipgrams which capture word sequences that contain (or skip over) arbitrary gaps, are the most informative features. [5] presented a corpus generated from review pairs on Amazon that can be used to identify sarcasm and irony in a tweet. [9] collected and annotated a set of ironic examples from a common collective Italian blog.

3 Corpus Creation and Annotation

In this section we explain the scheme used for corpus creation and annotation.

3.1 Corpus Creation

We constructed the Hindi-English code-mixed corpus using the tweets posted online since 2010. Tweets were scrapped from Twitter using the Twitter Python API which uses the advanced search option of twitter. We have mined the tweets using #irony, keywords 'irony' and 'ironic' and various hashtags from politics, sports and entertainment. The last three topics majorly but not essentially represent non-ironic tweets. As it is evident from example **T3** in section 1, it is not compulsory that irony is detected in all the tweets consisting of irony keywords and hashtags. We retrieved 1,19,885 tweets from Twitter in json format, which consists of information such as timestamp, URL, text, user, re-tweets, replies, full name, id and likes. An extensive semi-automated processing was carried out to remove all the noisy tweets. Noisy tweets are the ones which comprise only of hashtags or urls. Also, tweets in which language other than Hindi or English is used were also considered as noisy and hence removed from the corpus. Furthermore, all those tweets which were written either in pure English or pure Hindi language were removed, and thus, keeping only the code-mixed tweets. As a result, a dataset of 3055 code-mixed tweets was created. Newly created corpus and code is available online at Github.[1]

[1] https://github.com/deepanshu1995/Irony-Detection-Hindi-English-Code-Mixed-

3.2 Annotation

Annotation of the corpus was carried out as follows:

Language at Word Level : For each word, a tag was assigned to its source language. Three kinds of tags namely, 'eng', 'hin' and 'other' were assigned to the words by bilingual speakers. 'eng' tag was assigned to words which are present in English vocabulary, such as "Amazing", "Death", etc. 'hin' tag was assigned to Hindi words such as "sapna" (Dream), "hakikat" (Reality). The tag 'other' was given to symbols, emoticons, punctuations, named entities, acronyms, and URLs.

Ironic or Non-Ironic: : An instance of annotation is illustrated in figure 1. Each tweet is enclosed within <tweet></tweet>tags. First line in every annotation consists of tweet id. Language tags are added before every token of the tweet, enclosed within <word></word>tags. Each tweet is annotated with one of the two tags (Ironic or Non-Ironic). Irony is detected in 782 tweets. Remaining 2273 code-mixed tweets do not contain irony. The annotated dataset (consisting of tweet id's and annotated tag) with the classification system will be made available online later.

```
<tweet>
<id>831486289048457216<\id>
<word lang="eng">What</word>
<word lang="eng">an</word>
<word lang="eng">irony</word>
<word lang="other">?</word>
<word lang="hin">Jab</word>
<word lang="eng">relationship</word>
<word lang="hin">nai</word>
<word lang="hin">kiya</word>
<word lang="hin">tab</word>
<word lang="hin">sab</word>
<word lang="hin">Kuch</word>
<word lang="hin">mila</word>
<word lang="hin">Jab</word>
<word lang="eng">relationship</word>
<word lang="hin">mein</word>
<word lang="hin">hain</word>
<word lang="hin">tho</word>
<word lang="hin">Ek</word>
<word lang="hin">pic</word>
<word lang="hin">bhi</word>
<word lang="hin">nai</word>
<word lang="hin">mili</word>
<word lang="other">#ParSh</word>
<word lang="eng">Tales</word>
</tweet>
<class>
Ironic
</class>
```

Fig. 1. Annotated Instance

3.3 Inter Annotator Agreement

Annotation of the dataset to detect presence of irony was carried out by two human annotators having linguistic background and proficiency in both Hindi and English. A sample annotation set consisting of 50 tweets (25 ironic and 25 non-ironic) selected randomly from all across the corpus was provided to both the annotators in order to have a reference baseline so as to differentiate between ironic and non ironic text. In order to validate the quality of annotation, we calculated the inter-annotator agreement (IAA) for irony annotation between the two annotation sets of 3055 code-mixed tweets using Cohen's Kappa coefficient. Kappa score is 0.832 which indicates that the quality of the annotation and presented schema is productive.

4 System Architecture

In this section, we present our machine learning model for detecting irony in the code-mixed dataset described in the previous sections.

4.1 Pre-processing

Pre-processing of the code mixed tweets is carried out as follows. All the links and URLs are replaced with "URL". Tweets often contain mentions which are directed towards certain users. We replaced all such mentions with "USER". All the hashtags in the dataset are removed. All the emoticons used in the tweets are first stored to be used as a feature and then replaced with "Emoticon". All the punctuation marks in a tweet are removed. However, before removing them we store the count of each punctuation mark since we use them as one of the features in classification.

4.2 Classification Features :

In our work, we have used the following feature vectors to train our supervised machine learning model.

1. **Character N-Grams :** Character N-Grams are language independent and have proven to be very efficient for classifying text. These are also useful in the situation when text suffers from misspelling errors [10, 17, 20]. Group of characters can help in capturing semantic meaning, especially in the code-mixed language where there is an informal use of words, which vary significantly from the standard Hindi and English words. We use character n-grams as one of the features, where n vary from 1 to 3.
2. **Word N-Grams :** Bag of words feature is vital to capture the content in the text. Thus we use word n-grams, where n vary from 1 to 3 as a feature to train our classification models.

3. **Laugh Words and Emoticons :** Instead of using many exclamation marks internet users may use the sequence 'lmao' (i.e. laughing my ass of) or 'lol' (i.e. laughing out loud) or type hahaha. So we use a feature called laugh words which is the sum of all the internet laughs, such as 'haha', 'lol', 'lmao', 'rofl', 'lel', 'hehehe'. We also use emoticons as a feature for irony detection since they often represent textual portrayals of a writer's emotion in the form of symbols. We took a list of Western Emoticons from Wikipedia.[2]
4. **Punctuations :** Users often use exclamation marks when they want to express strong feelings. We count the occurrence of each punctuation mark in a sentence and use them as a feature.
5. **Intensifiers :** Users often tend to use intensifiers for laying emphasis on their feeling. A list of intensifiers was taken from Wikipedia. We count the number of intensifiers in a tweet and use the count as a feature.
6. **Negation words :** A list of negation words was taken from Christopher Pott's sentiment tutorial.[3] We count the number of negations in a tweet and use the count as a feature.
7. **Structure :** Ironical tweets in our dataset are often longer than other tweets. To capture this structure we use a group of features. (i) Number of characters present in the tweet. (ii) Number of words in the tweet. (iii) Average word length in the tweet.

Table 1. F1 Score for each feature using SVM classifier.

Features	F1 Score
All Features	0.77
Structural Features	0.64
Char N-Grams	0.77
Word N-Grams	0.70
Laugh Words + Emoticons	0.63
Punctuation Marks	0.63
Intensifiers	0.63
Negation Words	0.63

5 Experiments and Results

We performed experiments with two different classifiers namely Support Vector Machines with radial basis function kernel and Random Forest Classifier. Since the size of feature vectors formed are very large, we applied chi-square feature

[2] https://en.wikipedia.org/wiki/List_of_emoticons
[3] http://sentiment.christopherpotts.net/lingstruc.html

Table 2. F1 Score for each feature using Random Forest classifier.

Features	F1 Score
All Features	0.72
Structural Features	0.65
Char N-Grams	0.72
Word N-Grams	0.72
Laugh Words + Emoticons	0.63
Punctuation Marks	0.67
Intensifiers	0.63
Negation Words	0.63

selection algorithm which reduces the size of our feature vector to 1400[4]. For training our system classifier, we have used Scikit-learn [7]. In all the experiments, we carried out 10-fold cross validation. Table 1 and Table 2 describe the F1 score of each feature along with the F1 score when all features are used, in the case of Support vector machine and Random forest classifier respectively. Support vector machine performs better than Random forest classifier and gives a highest F1 score of 0.77 when all features are used. Character N-Grams proved to be most efficient in SVM, while word n-grams and character n-grams both resulted in best F1 score in the case of Random Forest Classifier.

6 Conclusion and Future Work

In this paper, we present an annotated corpus of Hindi-English code-mixed text, consisting of tweet ids and the corresponding annotations, which will be made freely available online later. We also present a supervised system used for detecting irony in the code-mixed text. The corpus consists of 3055 code-mixed tweets annotated as ironic or non-ironic. The features used in our classification system are character n-grams, word n-grams, emoticons, laugh words, punctuations, intensifiers and structural features. Best F1 score of 0.77 is achieved when all the features are incorporated in the feature vector using SVM as the classification system.
As a part of future work, the corpus can be annotated with part-of-speech tags at word level which could yield better results. Moreover, the annotations and experiments described in this paper can also be carried out for code-mixed texts containing more than two languages from multilingual societies, in future.

References

1. Aditya Joshi, Ameya Prabhu, Manish Shrivastava, and Vasudeva Varma: Towards Sub-Word Level Compositions for Sentiment Analysis of Hindi-English Code Mixed

[4] The size of feature vector was decided after empirical fine tuning

Text. In Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 2482-2491. 2016.
2. Antonio Reyes, Paolo Rosso, and Tony Veale: A multidimensional approach for detecting irony in twitter. Language resources and evaluation 47, no. 1 (2013): 239-268.
3. Arnav Sharma, Sakshi Gupta, Raveesh Motlani, Piyush Bansal, Manish Srivastava, Radhika Mamidi, and Dipti M. Sharma: Shallow parsing pipeline for hindi-english code-mixed social media text. arXiv preprint arXiv:1604.03136 (2016).
4. Carol Myers-Scotton: Dueling Languages: Grammatical Structure in Code-Switching. Claredon. (1993).
5. Elena Filatova: Irony and Sarcasm: Corpus Generation and Analysis Using Crowdsourcing. In LREC, pp. 392-398. 2012.
6. Erik Forslid and Niklas Wikén. Automatic irony-and sarcasm detection in Social media. (2015).
7. Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel et al: Scikit-learn: Machine learning in Python. Journal of machine learning research 12, no. Oct (2011): 2825-2830.
8. Francesco Barbieri, Francesco Ronzano, and Horacio Saggion. "Italian irony detection in twitter: a first approach." In The First Italian Conference on Computational Linguistics CLiC-it, p. 28. 2014.
9. Gianti Andrea, Bosco Cristina, Bolioli Andrea, and Luigi Di Caro. "Annotating irony in a novel italian corpus for sentiment analysis." In 4th International Workshop on Corpora for Research on EMOTION SENTIMENT & SOCIAL SIGNALS ES 2012, pp. 1-7. ELRA, 2012.
10. Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins: Text classification using string kernels. Journal of Machine Learning Research 2, no. Feb (2002): 419-444.
11. Kalika Bali, Jatin Sharma, Monojit Choudhury, and Yogarshi Vyas: "I am borrowing ya mixing?" An Analysis of English-Hindi Code Mixing in Facebook. In Proceedings of the First Workshop on Computational Approaches to Code Switching, pp. 116-126. 2014.
12. Khyathi Chandu Raghavi, Manoj Kumar Chinnakotla, and Manish Shrivastava: Answer ka type kya he?: Learning to classify questions in code-mixed language. In Proceedings of the 24th International Conference on World Wide Web, pp. 853-858. ACM, 2015.
13. Luisa Duran: Toward a better understanding of code switching and interlanguage in bilinguality: Implications for bilingual instruction. The Journal of Educational Issues of Language Minority Students 14, no. 2 (1994): 69-88.
14. Marjolein Gysels: French in urban Lubumbashi Swahili: Codeswitching, borrowing, or both?. Journal of Multilingual & Multicultural Development 13, no. 1-2 (1992): 41-55.
15. Pieter Muysken: Bilingual speech: A typology of code-mixing. Vol. 11. Cambridge University Press, 2000.
16. Souvick Ghosh, Satanu Ghosh, and Dipankar Das: Sentiment Identification in Code-Mixed Social Media Text. arXiv preprint arXiv:1707.01184 (2017).
17. Stephen Huffman. Acquaintance: Language-independent document categorization by n-grams. DEPARTMENT OF DEFENSE FORT GEORGE G MEADE MD, 1995.
18. Tony Vealy and Yanfen Hao: Detecting Ironic Intent in Creative Comparisons. In ECAI, vol. 215, pp. 765-770. 2010.

19. Utsab Barman, Amitava Das, Joachim Wagner, and Jennifer Foster: Code mixing: A challenge for language identification in the language of social media. In Proceedings of the first workshop on computational approaches to code switching, pp. 13-23. 2014.
20. William B. Cavnar, and John M. Trenkle: N-gram-based text categorization. Ann arbor mi 48113, no. 2 (1994): 161-175.
21. Yogarshi Vyas, Spandana Gella, Jatin Sharma, Kalika Bali, and Monojit Choudhury: Pos tagging of english-hindi code-mixed social media content. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 974-979. 2014.

Leveraging Cognitive Computing for Gender and Emotion Detection

Andrea Corriga[1], Simone Cusimano[1], Francesca M. Malloci[1], Lodovica Marchesi[1] and Diego Reforgiato Recupero[1]

Department of Mathematics and Computer Science,
University of Cagliari, Via Ospedale 72, 09124, Cagliari
me@andreacorriga.com,s.cusimano@studenti.unica.it,
francesca.mll@live.it,lodo.marchesi@gmail.com,
diego.reforgiato@unica.it

Abstract. In this paper we present a tool that performs two tasks: given an input image, first, (i) it detects whether the image corresponds to a male or female person and then (ii), it further recognizes which emotion the face expression of the detected person is conveying. We mapped the two tasks as multi-label classifications. The first one aims at identifying if the input image contains one of the following four categories: one male person, one female person, a group of both male and female persons or if the image does not contain any person. The second task is triggered whether the image has been recognized to belong to one of the first two classes and aims at detecting whether that image is conveying one among six different emotions: sadness, anger, surprise, happiness, disgust, fear. For both the problems, Microsoft Cognitive Services have been leveraged to extract tags from the input image. Tags are text elements that have been adopted to form the vectorial space model, using the bag of words model, that has been fed to the machine learning classifiers for the prediction tasks. For both tasks, we manually annotated 3000 images, which have been extracted from students who agreed using our system and providing their Facebook pictures for our analysis. Our evaluation uses Naive Bayes and Random Forest classifiers and with a 10-fold cross-validation reached satisfactory accuracies both for the two tasks and for the combination of them. Finally, our system works online and has been integrated with social media. In that way, any visitors logged in to Facebook through its APIs, is allowed to quickly classify any of their photos.

Keywords: Emotion Detection; Social Media; Cognitive Services; Image Classification; Machine Learning

1 Introduction

Today, around 2.45 billion people are active on different social media platforms where Facebook represents the one with the highest number of them (2 billions). Social media has become not only a key part of the modern lifestyle but also a

useful marketing channel for business of all sizes. Users uploads statuses, posts, images, and videos and, therefore, this contributes to the increase of available data in Internet [17]. Most of the time, this data is made accessible only by friends or friends of friends. What not so many people know is that all the information related to a certain user might be accessible by web applications that allow associating users' Facebook account to log in within their restricted area [4]. In fact, when visiting certain websites (e.g. news), there is often the possibility to associate user's Facebook account through a dedicated pop-up, avoiding the long times needed by the registration process. It usually takes one click to perform this operation as Facebook is usually left open in users browsers. This is allowed by Facebook APIs and a validation process of the application that is being developed and, potentially, gives to the website creators access to all the information of the users. This information cannot be published as it is for privacy reasons, but it can be processed and used to tune algorithms and systems. Processing all this amount of 'big data' might be expensive but today, thanks to the development of cognitive computing tools, the elaboration (of both text and images) can be fast enough. Big data offers plenty of opportunities to unlock novel insights from the huge amount of data that is available today. Although more data (e.g. text, images, videos, sensor data) is available than ever, only a small portion of it is being analyzed and used.

Understanding emotions [2] from posts and photos is a direction that many social networks are heading: emotional disclosure can foster interpersonal connectedness and individuals are motivated to express their emotions to maintain their relatedness to others. Moreover, analyzing pictures from social media and photo-sharing websites such as Flickr, Twitter, Tumblr, Facebook can give insights into the general sentiment of people about a given event, person, organization, etc. Recent works have already investigated the mechanisms of how social network structure influences the need for emotional expression [18]. On the other hand, if we can understand the emotion an image conveys we would be able to predict emotional tags on them. There are several works that aimed at using image processing techniques for such purpose[23, 25] but not so many that solve the problem by using text description extracted from the images.

Cognitive computing systems [14, 15] are emerging and represent the third era of computing. They have been used to improve several tasks which range from sentiment analysis [24, 22], to multi-class classification of e-learning videos [5], to classification of complaints in the insurance industry [13]. Cognitive computing systems rely on deep learning algorithms and neural networks to elaborate information by learning from a training set of data. They are perfectly tailored to integrate and analyze the huge amount of data that is being released today and available. Two very well known cognitive computing systems are IBM Watson[1] and Microsoft Cognitive Services[2]. They have been employed in several domains especially within life sciences research [3].

[1] `https://www.ibm.com/watson/`
[2] `https://azure.microsoft.com/en-us/services/cognitive-services/`

In this paper we present two multi-classification problems. The first one aims at detecting whether a given image contains one male person, or one female person, or two or more people of different sex or no person. If an image belongs to one of the first two classes (is a man or a woman) one more task is performed that aims at classifying the face expression of the input image according to six possible emotions: sadness, anger, surprise, happiness, disgust, fear. Microsoft Cognitive Services have been leveraged to extract textual tags from training and test set images and create a vectorial space using bag of word model (we employed term frequency and TF-IDF) that has been fed to different classification algorithms. Our system has been validated on 3000 images manually annotated and extracted from students Facebook profiles who agreed to have their Facebook pictures used for our analysis. A 10-fold cross-validation method has been performed. We obtained an accuracy of 65% for the first classification task, an accuracy of 60% for the second task and an accuracy of 56% for the combined task. They system has also been developed to work online and allows any user to associate his/her Facebook account for logging in and classify any of the user's photo through the two tasks. The remainder of this paper is organized as it follows: Section 2 includes related work. A brief description of the Microsoft Cognitive Services is given in Section 3. Section 4 describes the used dataset and how we turned the input images into text representation. Section 5 includes the adopted methodology and the performance evaluation we carried out. Section 6 gives some technical details of our developed system. Finally, Section 7 draws conclusions and directions where we are headed.

2 Related Work

Gender recognition is a domain that has attracted interest in both fundamental and applied research. It has mostly been targeted using computer vision algorithms and its resolution is still challenging. The difficulties emerge from the different positions of a face whose capture depends on the image acquisition process, and the intrinsic differences between people's faces [19]. We turned this problem in a text classification and, to do that, we employed Cognitive Computing Systems to detect textual descriptions of the input images. The other problem we address in this paper is the Emotion detection. This problem can be tackled with computer vision techniques if we want to extract facial expression from a given image [16] or Natural Language Processing and Semantic Web techniques if we are in presence of text and want to detect or extract emotions from it [24]. In literature several works have started using Cognitive Computing Systems to extract textual (syntactic and semantic) elements that have been used to generate the vectorial space using augmentation and/or replacement techniques [5, 7, 6, 11, 12]. Results have shown to improve baselines not using such features. The opportunity that such systems offer has been therefore exploited by several approaches, also within the Sentiment Analysis domain [10, 9], that improved their accuracy and raised the competitiveness of known Sentiment Analysis challenges [8, 20, 21].

3 Microsoft Cognitive Services

We employed the Computer Vision APIs (v1.0)[3]. They provide state of the art algorithms to process images. They can be used to determine if an image contains mature content, or estimating dominant and accent colors and so on. When performing a request, the input consists of an image URL. Within the request, there is an optional parameter used to specify which features to return (image categories are returned by default). The response will be returned in JSON. Metadata we collected are: *categories, tags, description, faces, image type, color,* and *adult*. The subscription we used is free and, as such, we were able to perform only 20 requests per minute. For this reason, the online application might take several minutes to retrieve the metadata of all the images of the user. On the other hand, the Emotion APIs take a facial expression in a given input image, and returns the confidence across a set of emotions. The detected emotions are anger, contempt, disgust, fear, happiness, neutral, sadness and surprise.

4 Dataset

Our dataset set is represented by 3000 images extracted from Facebook accounts of students at University of Cagliari who agreed participating to our analysis. A subset of 1000 images contained pictures of single persons and, therefore, were used for the evaluation of the second task. These images represented face expressions having six emotions: sadness, anger, surprise, happiness, disgust, fear. The first task has been evaluated on the entire collection of images. Tables 1 and 2 shows some statistics of the collected images dataset. Moreover, images have been first sent to Microsoft Cognitive Services to retrieve the tags and manually classified according to the two classifications: one label out of four classes for the gender recognition task and one more label out of six for the emotion detection class. For the emotion detection class we followed the same procedure of the generation of the Microsoft FER dataset [1]. During the annotation, all the tags indicating the gender of a person (e.g. lady, man, woman, boy, girl) have been removed. Finally, each image has been replaced by its corresponding textual representation using tags returned by Microsoft Cognitive Services.

Table 1. Statistics for the gender detection dataset.

Images of single male persons	449
Images of single female persons	551
Images of more than one person	976
No persons	1024

[3] `https://bit.ly/2KLkskV`

Table 2. Statistics for the emotion detection dataset.

sadness	145
anger	181
surprise	166
happiness	176
disgust	180
fear	152

5 Evaluation

As mentioned within Section 4, we leveraged the textual tags extracted from Microsoft Cognitive Services to create our vectorial space model using bag of word and TF and TF-IDF models. Tags are text elements returned for each image based on more than 2000 recognizable objects, living beings, scenery, and actions. Tags are not organized as a taxonomy and no inheritance hierarchies exist. Naive Bayes and Random Forest classifiers have been employed for the obtained accuracy shown in Table 3 using 10-cross validation technique.

Table 3. Accuracies for TF and TF-IDF for the gender recognition task.

Classifier	TF	TF-IDF
Naive Bayes	65%	63%
Random Forest	60%	58%

For the second task, again, we generated the vectorial space similarly as performed above for the first task. We tested the same classifiers and in Table 4 results of the obtained accuracy are shown using 10-cross validation technique.

Table 4. Accuracies for TF and TF-IDF for the emotion detection task.

Classifier	TF	TF-IDF
Naive Bayes	62%	59%
Random Forest	58%	55%

Finally, Table 5 shows the accuracy for the combination of the two tasks: given an image, it detects if it represents a single person and, in such a case, classifies which emotion it is conveying. For simplicity, the combined task has been represented as a multi-class classification problem with 8 categories: face expression of a single person conveying each of the six emotions, more than a single person, no persons and it has been tested on the entire collection of 3000 annotated images using 10-cross validation technique.

Table 5. Accuracies for TF and TF-IDF for the combination of gender recognition and emotion detection tasks.

Classifier	TF	TF-IDF
Naive Bayes	56%	55%
Random Forest	55%	54%

The accuracy has been computed according to the following equation;

$$accuracy = \frac{tp + tn}{tp + tn + fp + fn}$$

where tp, tn, fp and fn are, respectively, true positives, true negatives, false positives and false negatives.

Table 6 shows the confusion table for the first class (male) of the gender recognition task with the quantities above (true positive, etc.) properly indicated for the computation of the accuracy. Three more accuracies values have been computed using three more similar confusion tables for the remaining three classes. The overall accuracy has been calculated averaging the four accuracies thus computed. A similar process has been used to calculate the accuracy for the emotion detection task and the combined task.

		Actual class	
		Male person.	*Non Male person*
Predicted class	*Male person*	True Positives	False Positives
	Non Male person	False Negatives	True Negatives

Table 6. Confusion table for the gender recognition task.

6 System Workflow

In this paragraph we will describe how our system works online. It is a web application composed by two main modules: the first is a web application developed using CodeIgniter Web Framework[4] which allows building robust solutions; the second module is a Python HTTP Apache Server.

We used the Facebook Graph API[5] by obtaining first the API KEY where we had to specify the needed security scopes for our application. We explicitly mentioned *user_photos* to gain the URLs of the users' public photos. During this step we had to submit a short description of our project for the Facebook team to validate. Once validated, we were able to extract the list of the public images of the user navigating our web application. Figure 1 shows the architecture of the first task[6]. A certain user visits our web application and he is redirected to

[4] `https://codeigniter.com/`
[5] `https://developers.facebook.com/docs/graph-api/`
[6] `https://bit.ly/2IrDzC1`

Facebook.com to start the authentication flow. After a successful log in, the user is redirected back to our server. The public images of the user are retrieved using a valid authentication token. The photos are sent to the Cognitive Services to fetch the related metadata. The images are shown to the user. Each photo can be clicked and a pop up appears with the related metadata and with one of the four categories related to the first task we predict on the fly with our trained classifier. In the background, a new JSON file for that user is created containing his/her images with the associated metadata and the predicted class and stored on HDFS through its REST APIs.

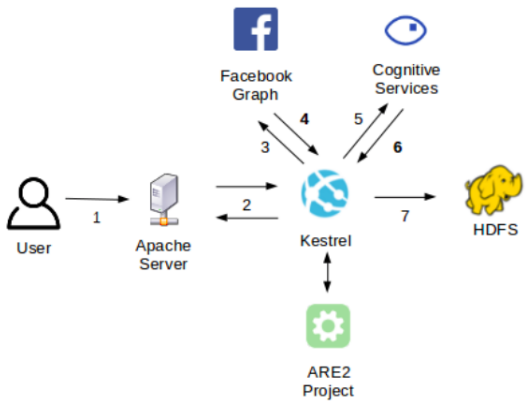

Fig. 1. Architecture of the first task.

Figure 2 shows the architecture of the second task[7] which can be run in standalone mode or once the first task has recognized the input image as a single person. In the first case, the user can decide whether to classify the image by logging in with his/her user profile through the Facebook entry or by uploading the image through the upload button. After one or more images have been chosen, a task in background starts to send the image to the Computer Vision Services and Emotion Services to fetch the related metadata. The description tags returned from Computer Vision are sent to our classification to predict the corresponding emotion. Once the computation has been completed, the user can see for that image the results of our classification related to the emotion conveyed by the face expression extracted from the input image.

7 Conclusions

In this paper we presented an approach that performs two multi-class classification tasks: detecting gender of a given image and detecting emotions of face

[7] `https://bit.ly/2jQpOzb`

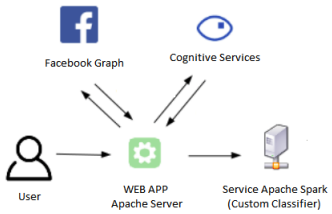

Fig. 2. Architecture of the second task.

expression for images detected within the first task. The system has been developed and works on-line and integrated with Facebook so that visitors can log in associating their Facebook account and deciding to perform the classification task on their images. For each selected image, Microsoft Cognitive Services are called and image tags collected in order to create a vectorial space representation of the input image. Our evaluation on 3000 images for the first task and 1000 images for the second task using two different classifiers with 10-cross validation generated accuracy respectively of 65% using TF and 63% using TF-IDF for the first task, 59% using TF and 55% using TF-IDF for the second task and, 56% (TF) and 55% (TF-IDF) for the combined task. As future work we are already collecting more images using crowd sourcing tools and will apply deep learning models on machines with fast NVIDIA GPUs (e.g. TitanX).

Acknowledgments

The authors gratefully acknowledge Sardinia Regional Government for the financial support (Convenzione triennale tra la Fondazione di Sardegna e gli Atenei Sardi Regione Sardegna - L.R. 7/2007 annualità 2016 - DGR 28/21 del 17.05.201, CUP: F72F16003030002).

References

1. Emad Barsoum, Cha Zhang, Cristian Canton-Ferrer, and Zhengyou Zhang. Training deep networks for facial expression recognition with crowd-sourced label distribution. *CoRR*, abs/1608.01041, 2016.
2. Davide Buscaldi and Belém Priego Sanchez. LIPN-UAM at emoint-2017: Combination of lexicon-based features and sentence-level vector representations for emotion intensity determination. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, WASSA@EMNLP 2017, Copenhagen, Denmark, September 8, 2017*, pages 255–258, 2017.
3. Ying Chen, JD Elenee Argentinis, and Griff Weber. Ibm watson: How cognitive computing can be applied to big data challenges in life sciences research. *Clinical Therapeutics*, 38(4):688 – 701, 2016.

4. Gianpiero Costantino, Fabio Martinelli, and Daniele Sgandurra. Are photos on social networks really private? In *2013 International Conference on Collaboration Technologies and Systems, CTS 2013, San Diego, CA, USA, May 20-24, 2013*, pages 162–165, 2013.
5. D. Dessì, G. Fenu, M. Marras, and D. Reforgiato Recupero. Leveraging cognitive computing for multi-class classification of e-learning videos. In E. Blomqvist, K. Hose, H. Paulheim, A. Ławrynowicz, F. Ciravegna, and O. Hartig, editors, *The Semantic Web: ESWC 2017 Satellite Events*, pages 21–25, Cham, 2017. Springer International Publishing.
6. D. Dessì, G. Fenu, M. Marras, and D. Reforgiato Recupero. Bridging learning analytics and cognitive computing for big data classification in micro-learning video collections. *Computers in Human Behavior*, 2018. Article in Press.
7. D. Dessì, D. Reforgiato Recupero, G. Fenu, and S. Consoli. Exploiting cognitive computing and frame semantic features for biomedical document clustering. In *Proc. of SeWeBMeDA@ESWC 2017, Portoroz, Slovenia, May 28, 2017.*, pages 20–34, 2017.
8. M. Dragoni and D. Reforgiato Recupero. Challenge on fine-grained sentiment analysis within eswc2016. *Communications in Computer and Information Science*, 641:79–94, 2016.
9. Mauro Dragoni and Giulio Petrucci. A neural word embeddings approach for multi-domain sentiment analysis. *IEEE Trans. Affective Computing 8(4)*, pages 457–470, 2017.
10. Mauro Dragoni and Giulio Petrucci. A fuzzy-based strategy for multi-domain sentiment analysis. *International Journal of Approximate Reasoning*, 93:59–73, 2018.
11. Amna Dridi, Mattia Atzeni, and Diego Reforgiato Recupero. Finenews: fine-grained semantic sentiment analysis on financial microblogs and news. *International Journal of Machine Learning and Cybernetics*, 2018.
12. Amna Dridi and Diego Reforgiato Recupero. Leveraging semantics for sentiment polarity detection in social media. *International Journal of Machine Learning and Cybernetics*, 2017.
13. J Forster and B Entrup. A cognitive computing approach for classification of complaints in the insurance industry. *IOP Conference Series: Materials Science and Engineering*, 261(1):012016, 2017.
14. J. O. Gutierrez-Garcia and E. López-Neri. Cognitive computing: A brief survey and open research challenges. In *2015 3rd International Conference on Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence*, pages 328–333, 2015.
15. John E. Kelly and Steve Hamm. *Smart Machines: IBM's Watson and the Era of Cognitive Computing.* Columbia University Press, New York, NY, USA, 2013.
16. Jyoti Kumari, R. Rajesh, and K. M. Pooja. Facial expression recognition: A survey. *Procedia Computer Science*, 58:486–491, 2015.
17. Savita Kumari. Impact of big data and social media on society. *Global Journal For Research Analysis*, 5, March 2016.
18. Han Lin, William Tov, and Lin Qiu. Emotional disclosure on social networking sites. *Comput. Hum. Behav.*, 41(C):342–350, December 2014.
19. Choon-Boon Ng, Yong-Haur Tay, and Bok-Min Goi. A review of facial gender recognition. *Pattern Analysis and Applications*, 18(4):739–755, Nov 2015.
20. D.R. Recupero, M. Dragoni, and V. Presutti. Eswc 15 challenge on concept-level sentiment analysis. *Communications in Computer and Information Science*, 548:211–222, 2015.

21. D. Reforgiato Recupero, E. Cambria, and E. Di Rosa. Semantic sentiment analysis challenge at eswc2017. *Communications in Computer and Information Science*, 769:109–123, 2017.
22. Diego Reforgiato Recupero, Valentina Presutti, Sergio Consoli, Aldo Gangemi, and Andrea Giovanni Nuzzolese. Sentilo: Frame-based sentiment analysis. *Cognitive Computation*, 7(2):211–225, 2015.
23. Can Xu, Suleyman Cetintas, Kuang-Chih Lee, and Li-Jia Li. Visual sentiment prediction with deep convolutional neural networks. *CoRR*, abs/1411.5731, 2014.
24. Ali Yadollahi, Ameneh Gholipour Shahraki, and Osmar R. Zaiane. Current state of text sentiment analysis from opinion to emotion mining. *ACM Comput. Surv.*, 50(2):25:1–25:33, May 2017.
25. Quanzeng You, Jiebo Luo, Hailin Jin, and Jianchao Yang. Robust image sentiment analysis using progressively trained and domain transferred deep networks. *CoRR*, abs/1509.06041, 2015.

In Search for Lost Emotions: Deep Learning for Opinion Taxonomy Induction

Elena Melnikova[1], Emmanuelle Dusserre[2] and Muntsa Padró[2]

[1] Innoradiant, Meylan 38240, France
[2] Eloquant, Gières 38610, France
elena.melnikova@innoradiant.com
emmanuelle.dusserre@eloquant.com
muntsa.padro@eloquant.com

Abstract. In this article, we present an approach for using word2vec to automatically enrich the opinions' taxonomy used by a sentiment analysis system. More specifically, we worked on emotion lexicon for the field of customer relationship management. The proposed method consists of searching for the nearest distributional neighbors of each source word, and add them to the lexicon of emotions. The hypothesis is that the contextual neighbors of the emotions will also carry an emotional coloring. The results of this experiment show that the neighborhood lexicon is not sufficiently representative. Nevertheless, most of the collected items seem to express another type of opinion, namely judgments. This unexpected result allows us to broaden our taxonomy of opinions with a new informational field, richer and more expressive.

Keywords: emotions, judgments, opinions, taxonomy, word2vec, deep learning, sentiment analysis.

1 Introduction

The automatic analysis of customer opinions is becoming one of the most pervasive concerns of companies studying customer reviews. The sentiments or opinions expressed in these reviews are important indicators for the company's decision-making strategy. Thus, sentiment analysis (SA) systems need to be reliable and constantly updated.

Many SA systems are often based on social networks, tweets or SMS corpora [6, 1, 3]. The analysis of opinions essentially focuses on polarity detection in customers feedbacks (positive, negative or neutral), [4]. Our SA system for French [7] performs the extraction of different kind of fine-grained opinions, including emotions, to extract more detailed information than just positive vs negative polarity. This fine-grained detection is performed using a taxonomy associating words or expressions to the classes to be detected [11]. Though, building a complete taxonomy can be very time consuming, and the relevant terms might depend on the domain.

The present work is motivated by the wish to semi-automatically enrich the taxonomy, in order to achieve a greater accuracy and easily adapt the system to different sub-domains. To do so, we propose the use of word2vec [10] to add entries to an existing taxonomy. This method has been widely and successfully used in semantic

analysis and other NLP tasks [2, 9]. We assume that the matrix constructed by traversing our domain specific corpus (Customer Relationship Management or CRM) would locate close to each other emotional words belonging to the same class (from the distributional point of view). By adopting this method based on deep learning, we intend to check if it is relevant for the enrichment of our emotions taxonomy.

The rest of this paper is structured as follows. In Section 2, we describe the word2vec method and the procedure to enrich the emotion taxonomy. Section 3 reports experiments and result discussion. Finally, Section 4 concludes our work.

2 The word2vec method

Word2vec is a statistical language model based on neural networks developed by a team of researchers under the direction of T. Mikolov [10] at Google[1]. This method is used to produce word embeddings: words in a corpus are represented as vectors in a multidimensional space, their position in this space corresponding to their semantic representation [8, 10].

The word2vec technique is based on a distributional hypothesis: words that appear in the same context share semantic values, thus, words that are close in the space are semantically close. Our goal is to use this method to find sets of closest words (or related words) and assign them to the same semantic class to enrich other kind of taxonomies, as it was shown in [2, 5].

We followed a procedure which contains two major steps. The first one is to create a word2vec model from a given corpus. The model creation includes the processing of the corpus (tokenization, lemmatization, etc.) and the induction of the model from it. The second step is to use the model to calculate the distance between a selected word (*seed* word) and the other words in the corpus. In our work, the words already present in the taxonomy (that already have an assigned semantic tag) serve as seed words. The objective is to assign to the closest words of the seed words the same semantic class as them. The distance between two words is calculated with the cosine of the angle between the vectors that represent them. The more this cosine is close to 1, the closest the neighbor is to the source word.

3 Experiments and results

We first focused on the improvement of the emotion detection module, by reviewing the emotions taxonomy used by our system. This taxonomy contains lexical items from different linguistic genres[2] (literature, psychology, familiar) with 41 classes and more than 1100 words. This seemed too large and it was not adapted to our CRM domain. Thus, the taxonomy was considerably reduced to constitute a specific sub-taxonomy (10 classes, 360 words). Among these classes, we find: ANGER, SADNESS, DISSATISFACTION, LIKING, SATISFACTION, DISLIKE, DOUBT, TRUST, CALMNESS. The lexicon for this reduced classification seemed quantitatively "poor" and not adequate to the CRM domain. Thereby, we found it necessary to increase the number of words

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

[2] WordNet based taxonomy : https://wordnet.princeton.edu/wordnet/frequently-asked-questions/database/ (Princeton University 2018)

for certain classes that contained less than 10 items (SATISFACTION, TRUST, DISTRUST, DOUBT, DISLIKE, CALMNESS).

The corpus used for the model has 15 million words and it is very specific to the CRM domain. Despite word2vec models are expected to work better with bigger corpus, [5] showed that for specific domains it is preferable to have a domain specific corpus than huge amount of data.

Table 1 shows a sample of results obtained when applying the word2vec method for a selection of words of emotion classes poorly endowed. The headline shows the seed words with their emotion classes and the following lines, the neighbors proposed with their cosine.

Table 1**.** Extraction of the nearest neighbors

Confiance (trust) **class: TRUST**		***Satisfait*** (satisfied) **class: SATISFACTION**		***Doute*** (doubt) **class: DOUBT**	
Neighbours	**Cos.**	**Neighbours**	**Cos.**	**Neighbours**	**Cos.**
perdre (to lose)	0.46	*résilier* (to cancel)	0.31	*red* (red)	0.39
préférer (to prefer)	0.41	*resolu* (resolved)	0.31	*prévenir* (to prevent)	0.37
abus (breach)	0.39	*accueil* (reception)	0.32	*accord* (agreement)	0.38
quitter (to leave)	0.30	*satisfaire* (to satisfy)	0.32	*considération* (consideration)	0.34

This extract is obtained by using a threshold of 0.3, meaning that only words with a cosine bigger than 0.3 are suggested as candidates to be added to the taxonomy. The threshold is set heuristically, looking for a compromise where we retrieve enough candidates without too much noise. In the obtained results we noticed different types of distributional neighborhoods:

1. collocations (*perdre confiance* (lose confidence), *abus de confiance* (breach of trust), *gagner en (la) confiance (de qqn)* (gain someone's trust), *satisfait de l'accueil* (satisfied with the reception));
2. synonyms, antonyms and derived forms (*satisfaire* (to satisfy) for *satisfait* (satisfied));
3. neighbors of a different semantic tag (*code* (code), *œuvre* (work), *cordialement* (cordially), *fixer* (to fix), *supprimer* (to delete), etc.).

The lexica that we expected to find should come from the second type of neighborhood, according to our goal and hypothesis. The obtained results show that these lexica are rather infrequent and, most often, they contain antonyms. It means that we cannot increase the classes of our taxonomy by using the closest neighbors. Thus, our hypothesis of enriching the emotions taxonomy, and especially poorly endowed emotions classes by using word2vec is not confirmed. Nevertheless, the idea of resorting to a new method which provides a rich lexical and statistical information, seems to us very attractive and less exploited.

3.1 The unexpected result

To better understand the results, we study more closely the lexica coming from the most numerous type of neighborhood, the neighbors with different semantic tag.

Some of those neighbors can be considered just noise, but we have distinguished in this lexical layer a category of words that could characterize non-emotional opinions, the *judgments*. For example, the words like *beau* (beautiful) and *accord* (agreement) are positive polarity judgments. The words *proche* (close adj.), *rapide* (fast) are positive or negative depending on the domain[3]. This type of polar judgment lexicon is as important as the emotion lexicon for the detection and analysis of customers opinions.

At the sight of these results, we extended the experiment by using word2vec with the whole taxonomy of emotions as seed words. This allowed us to identify more words that are likely to express judgments (see Table 2).

Table 2. Extraction of the nearest neighbors to search for the lexicon of judgments.

Déplaisant (unpleasant) **Class: DISLIKE**		***Rassurant*** (reassuring) **Class: TRUST**		***Satisfait*** (satisfied) **Class: SATISFACION**	
Neighbours	**Cos.**	**Neighbours**	**Cos.**	**Neighbours**	**Cos.**
Serviabilité (helpfulness)	0.28	***Comprehensible*** (understandable)	0.29	*Traitement* (proscessing)	0.23
Relation (relationship)	0.26	*Approprier* (to appropriate)	0.26	***Courtois*** (courteous)	0.23
Impeccable (faultless)	0.21	*Plateforme* (platform)	0.26	***Performant*** (performing)	0.20

Table 2 shows in bold the neighbors that are judgments and that can be added to our lexica[4]. In fact, our SA system already contains a little judgment lexicon (13 classes with 197 items). The new results can be gathered in a class APPRECIATION completing this lexicon. Note that the selection of this class for the judgment words is done manually, and not assigning the semantic class of the seed word, as it was our hypothesis. Even though it demands more human intervention, the possibility of enriching the judgment taxonomy seems to us a promising outcome of this work, since by studying the output of word2vec we can find new relevant classes. Word2vec finds more lexica related to judgments than emotions since CRM corpora contain much more of these lexica. This is also why judgments constitute key elements to be added to our SA system.

3.2 The extension of the judgments taxonomy

The judgments lexica extracted from CRM corpus better characterizes the CRM-specific classes (see Fig. 1). This idea is corroborated by a recent research study [5]. The augmented judgment taxonomy contains 18 classes and 261 items compared to 13 classes and 197 words from the initial taxonomy of judgments. Table 3 shows an extract from this ranking that should be expanded and completed.

[3] Their polarity reveals in context (*le personnel est proche du client* (the staff is close to the client*)* [positive judgment in commercial context vs negative judgment in familiar context]).

[4] In this second experiment we use 0.2 as threshold in order to obtain more candidates to be added to the judgment taxonomy

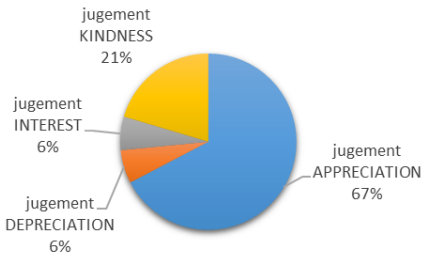

Fig. 1. The distribution of the lexicon of judgments extracted from CRM corpus.

Table 3. An extract from the new taxonomy of judgments.

Opinion's type	Class	Words
judgment	APPRECIATION	*Positive* (positive) *Apprécier* (to appreciate) *Méritoire* (meritorious)
	DEPRECIATION	*Déprécié* (depreciated) *Fichu* (damn) *Anormal* (unnatural)

4 Conclusion and Further Work

In this work, we have tested the applicability of word2vec to enrich an existent emotion taxonomy by finding semantically close words. The obtained results are not very satisfactory, since very few new emotional words are extracted. Nevertheless, the method allowed us to extract judgment words which are also very important for the SA system and much more frequent in our domain. Thus, we can use this new taxonomy as a resource in our system and we conclude that word2vec is a useful method to enrich existing taxonomies and even to discover new classes. But to guarantee the quality of final resources human intervention is needed. Table 4 summarizes the most important positive and negative points we spotted with our experiments with word2vec.

Table 4. Advantages and disadvantages of word2vec

Criteria	Advantages	Disadvantages
Speed	Tailor made extraction (depending on the threshold, PoS filter, etc.)	The learning time can take several tens of minutes depending on the memory size of the machine and corpus size.
Efficiency	The matrix can be trained with several types of corpus	Ambiguous lexica is often present, which requires to check its meaning in the context to associate it to a class or to remove it from the taxonomy
Reliability	The use of a specific domain corpus improves the results	We cannot predict the type of extracted lexica (expected affects, harvested judgments)

In the future, we plan to perform an extrinsic evaluation of the developed taxonomies. The convenience of using word2vec to enrich the taxonomies seems clear to us, when considering as an alternative the manual development of the taxonomies. Nevertheless, a final evaluation of our SA system before and after enriching the taxonomy needs to be done. For that, we are currently working on a CRM gold-standard. Also, we plan to perform the same experiments with word2vec trained on another CRM corpus to adapt the taxonomy of its sub-domain.

Furthermore, the taxonomy enriched with word2vec can also serve as input for a new run of the taxonomy enrichment system. Thus, we could perform a bootstrap to iteratively enrich the taxonomy of emotions and judgments.

References

1. Abdaoui, A., Nzali, M.D.T., Azé, J., Bringay, S., Lavergne, Ch., et al. ADVANSE: Sentiment, Opinion and Emotion Analysis in French Tweets. DEFT: Défi Fouille de Texte, Jun 2015, Caen, France. Actes de la 11e Défi Fouille de Texte (2015) <hal-01222629>
2. Baroni, M., Dinu, G. & Kruszewski, G. Don't count, predict! A systematic comparison of context-counting vs. contextpredicting semantic vectors. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 238–247, Baltimore, Maryland, June. Association for Computational Linguistics (2014).
3. Dini, L., Bittar, A., Robin, C., Segond, F., Montaner, M., SOMA: The Smart Social CRM. Handling Semantic Variability of Emotion Analysis with Hybrid Technologies. In Sentiment Analysis in Social Network Elsevier, chapter 13 (2016).
4. Dridi, A., Reforgiato Recupero, D. Leveraging semantics for sentiment polarity detection in social media. International Journal of Machine Learning and Cybernetics (2017).
5. Dusserre, E. & Padró, M. Bigger does not mean better ! We prefer specificity. In 12th International Conference on Computational Semantics (IWCS). 19-22 September 2017 Montpellier (France) (2017).
6. Hamon, T., Fraisse, A., Paroubek, P., Zweigenbaum, P., Grouin, C.. Analyse des émotions, sentiments et opinions exprimés dans les tweets: présentation et résultats de l'édition 2015 du défi fouille de texte (DEFT). In 22ème Traitement Automatique des Langues Naturelles (2015).
7. Maurel, S., Curtoni, P., & Dini, L. A hybrid method for sentiment analysis. In INFORSID. Présenté à Défi Fouille de Texte 2007 (DEFT'07) (2008).
8. Levy, O., Goldberg, Y. & Dagan, I: Improving Distributional Similarity with Lessons Learned from Word Embeddings. Transactions of the Association for Computational Linguistics (2015).
9. Maître, J., Menard, M., Chiron, G., Bouju, A . Utilisation conjointe LDA et Word2Vec dans un contexte d'investigation numérique. Extraction et Gestion des Connaissances 2017, Jan 2017, Grenoble, France (2017).
10. Mikolov T., Sutskever I., Chen K., Corrado G., Dean G. Distributed Representations of Words and Phrases and their Compositionality. NIPS'13 Proceedings of the 26th International Conference on Neural Information Processing Systems. Lake Tahoe, Nevada (2013).
11. Whitelaw, C., Garg, N., Argamon, S. Using Appraisal Taxonomies for Sentiment Analysis. Conference: The Second Midwest Computational Linguistic Colloquium, MCLC (2005).

Detecting Truthful and Useful Consumer Reviews for Products using Opinion Mining

Kalpana Algotar and Ajay Bansal

Arizona State University, Mesa AZ 85212 USA
{kalgotar, ajay.banssal}@asu.edu

Abstract. Individuals and organizations rely heavily on social media these days for consumer reviews in their decision-making on purchases. However, for personal gains such as profit or fame, people post fake reviews to promote or demote certain target products as well as to deceive the reader. To get genuine user experiences and opinions, there is a need to detect such spam or fake reviews. This paper presents a study that aims to detect truthful, useful reviews and ranks them. An effective supervised learning technique is proposed to detect truthful and useful reviews and rank them, using a 'deceptive' classifier, 'useful' classifier, and a 'ranking' model respectively. Deceptive and non-useful consumer reviews from online review communities such as amazon.com and Epinions.com are used. The proposed method first uses the 'deceptive' classifier to find truthful reviews followed by the 'useful' classifier to find whether a review is useful or not. Manually labeling individual reviews is very difficult and time consuming. We incorporate a dictionary that makes it easy to label reviews. We present the experimental results of our proposed approach using our dictionary with 'deceptive' classifier and 'useful' classifier.

Keywords: Text Classification, Spam Review Detection, Opinion Mining, Supervised Learning.

1 Introduction

Nowadays, consumers looking to buy a product increasingly rely on user-generated online reviews to make or reverse their purchase decisions. Positive reviews of a product greatly influence the person's decision to buy the product. However, if one sees many negative reviews, he/she will most likely choose a different product. The outcome of positive reviews gives significant profit and advertizing for the seller and their organization. This in turn creates a market for incentivizing opinion spam. This has resulted in more and more people trying to game the system by writing fake reviews to harm or promote some products or services. A fake review means that it is either a positive review written by the business owners themselves (or people they contract to write reviews) or a negative review written by a business's competitors. Those fake reviews try to deliberately mislead readers by giving fake reviews to some entities (e.g. products) in order to promote them or to damage their reputation.

Opinion spamming refers to writing fake reviews that try to deliberately mislead human readers. The focus of spam research in the context of online reviews has been primarily on detection. Cornell University has developed a model to spot fake, non-fake review for hotels [3] as well as some existing works have been done by other researchers to detect fake reviews and spam reviewers. Recent studies, however, show that opinion spam is not easily identified by human readers [9]. In particular, humans have a difficult time identifying deceptive messages from consumer reviews. We decided to work on the same issue for product by taking different approach to make the process easier. In this approach, we choose Cornell model [3] as a base to prepare our own dictionary for fake, non-fake reviews. Our, automated approach has emerged to reliably label reviews as truthful vs. deceptive as well as second approach to label useful vs. not-useful using reader's rating on consumer's review. We train SVM text classifier using a corpus of truthful and deceptive as well as useful and not-useful reviews from Amazon and Epinion. We applied our approach to the domain of camera reviews and present the results.

The rest of the document is organized as follows: Section 2 presents related work. Background material related to this project is presented in Section 3. Our proposed approach and its implementation is presented in Section 4. Section 5 presents the experiments and analysis followed by conclusions and future work in Section 6.

2 Related Work

Web spam and email spam have been investigated extensively. The objective of Web spam is to make search engines rank the target pages high in order to attract people to visit these pages. Web spam can be categorized into two main types: content spam and link spam. Link spam is spam on hyperlinks that are placed between pages, which does not exist in reviews as usually there are no links within them. Content spam tries to add irrelevant or remotely relevant words in target pages to fool search engines to rank the target pages high. Another related research is email spam [5, 8, 14], which is also quite different from review spam. Email spam usually refers to unsolicited commercial advertisements. Although this exists, advertisements in reviews are not as frequent as in emails. They are also relatively easy to detect. Deceptive opinion spam is much harder to deal with. We present below, different approaches taken opinion spam detection.

2.1 Review Spam Detection

A preliminary study was reported in [8] to study spam review and spam detection based on finding duplicates and classification. That study proposed to treat duplicate reviews as positive training examples (with label fake), and the rest of the reviews as the negative training examples (with label non-fake). For the rest of spam (fake) reviews, they detected based on 2-class classification (spam and non-spam). In addition, they found that 52% of the highly ranked non-duplicate reviews had more than 1800 words, much higher than the average length of a normal review, and were regarded as spam reviews. A more in-depth investigation was given in [6] where three types of spam review were identified, namely untruthful reviews (reviews that promote or demote products), reviews on brands but not products, and non-reviews (e.g., advertisements). By representing a review using a set of review, reviewer and product-level features, classification techniques were used to assign spam (fake) labels to

reviews. In particular, untruthful review detection is performed by using duplicate reviews as the positive training examples (fake) and the rest of the reviews as negative training examples (non-fake) and for rest of the types manual labeling was done. In [16] neural network based model was used for representation learning of reviews.

2.2 Reviewer Spam Detection

Some of the related research addresses the problem of review spammer detection, or finding users who are the source of spam reviews. Reviews usually come with ratings. Detecting unfair ratings has been studied in several works including [4, 10]. The techniques used include: (a) clustering ratings into unfairly high ratings and unfairly low ratings, and (b) using third party ratings on the producers of ratings and ratings from less reputable producers are then deemed as unfair. Once unfair ratings are found, they can be removed to restore a fair item evaluation system. These works did not address review spammer detection directly on the reviews. They usually did not conduct evaluation of their techniques on real data.

2.3 Helpful Review Detection and Prediction

Review helpfulness prediction is closely related to review spam detection described in above. A helpful review is one that is informative and useful to the readers. The purpose of predicting review helpfulness is to faciliate review sites to provide feedback to the review contributors and to help readers choose and read high quality reviews. A classification approach to solving helpfulness prediction using review content and meta-data features was developed in [7]. The meta-data features used are review's rating and the difference between the review rating and the average rating of all reviews of the product. Liu et. al proposes to derive features from reviews content that correspond to informativeness, readability, and subjectiveness aspects of the review [9]. These features are then used to train a review helpfulness classification method.

Amazon.com allows users to vote if a review is helpful or not. These helpfulness votes are manually assigned and are thus subjective and possibly abused. Danescu-Niculescu-Mizil et. al found that a strong correlation between proportion of helpful votes of reviews and the deviation of the review ratings from the average ratings of products [3]. This correlation illustrates that helpful votes are generally consistent with average ratings. The study is however conducted at the collection level and does not provide evidence to link spam and helpfulness votes. Ott and others [11] presented a framework for estimating the prevalence of deception in online review communities. In this task, they paid one US dollar ($1) to each of 400 unique Mechanical Turk workers to write a fake positive (5-star) review for one of the 20 most heavily-reviewed Chicago hotels on TripAdvisor. For consistency with labeled deceptive review data, they simply labeled as truthful all positive (5-star) reviews of the 20 previously chosen Chicago hotels.

Detecting spam and predicting helpfulness are two separate problems since not-useful reviews are not necessarily fake. A poorly written review may be not-useful but is not fake. Spam reviews usually target specific products while not-useful votes may be given to any products. Given the motive driven nature of spamming activities, review spam detection will therefore require an approach different from not-useful review detection. Our proposed technique aims to detect truthful, useful reviews and provide a ranking of the reviews.

3 Background

3.1 Supervised Learning Methods:

A computer system learns from training data that represents some "past experiences" of an application domain. In this section, we briefly describe the various classification methods used in order to categorize reviews into deceptive, truthful and useful, not-useful. Classification involves labeling of the data (observations, measurements) with pre-defined classes. We have used three supervised learning algorithms: Support Vector Machine, Naïve Bayes, and K-Nearest Neighbor.

Support Vector Machines:

Support Vector Machines [1] are supervised learning methods used for classification, as well as regression. The advantage of Support Vector Machines is that they can make use of certain kernels in order to transform the problem, such that we can apply linear classification techniques to non-linear data. Applying the kernel equations arranges the data instances in such a way within the multi-dimensional space, that there is a hyper-plane that separates data instances of one kind from those of another. The kernel equations may be any function that transforms the linearly non-separable data in one domain into another domain where the instances become linearly separable. Kernel equations may be linear, quadratic, Gaussian, or anything else that achieves this particular purpose. Once we manage to divide the data into two distinct categories, our aim is to get the best hyper-plane to separate the two types of instances. This hyper-plane is important because it decides the target variable value for future predictions. We should decide upon a hyper-plane that maximizes the margin between the support vectors on either side of the plane that is displayed in the Figure 1.

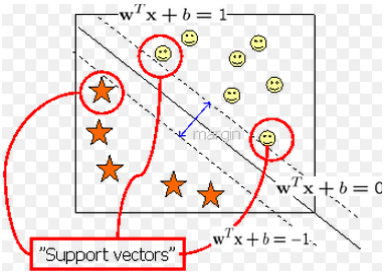

Fig. 1. Support Vector Machine

The data instances that were not linearly separable in the original domain have become linearly separable in the new domain, due to the application of a function (kernel) that transforms the position of the data points from one domain to another. This is the basic idea behind Support Vector Machines and their kernel techniques. Whenever a new instance is encountered in the original domain, the same kernel function is applied to this instance too, and its position in the new domain is found out. In our experiments too, it is seen that Support Vector Machines usually have the highest accuracy among any of the other classification methods.

Naïve Bayes Classifier:

The Naïve Bayes classifier [1] is based on the Bayes rule of conditional probability. It makes use of all the attributes contained in the data, and analyses them individually as

though they are equally important and independent of each other. For example, consider that the training data consists of various animals (for example: elephants, monkeys, and giraffes), and our classifier has to classify any new instance that it encounters. We know that elephants have attributes like they have a trunk, huge tusks, a short tail, are extremely big, etc. Monkeys are short in size, jump around a lot, and can climbing trees; whereas giraffes are tall, have a long neck and short ears.

The Naïve Bayes classifier will consider each of these attributes separately when classifying a new instance. So, when checking to see if the new instance is an elephant, the Naïve Bayes classifier will not check whether it has a trunk and has huge tusks and is large. Rather, it will separately check whether the new instance has a trunk, whether it has tusks, whether it is large, etc. It works under the assumption that one attribute works independently of the other attributes contained by the sample. In our experiments, it is seen that the Naïve Bayes classifier shows a drop in performance, when compared with K-NN and Support Vector Machines.

K-Nearest Neighbor:

The K-nearest neighbor [15] algorithm is a method for classifying objects based on closest training examples in the feature space. Unlike all the previous learning methods, K-NN doesn't build the model from the training data. No explicit model for the probability density of the classes is formed; each point is estimated locally from the surrounding points. The k-nearest-neighbor classifier is commonly based on the Euclidean distance between a test sample and the specified training samples. Given a test instance, a distance metric is computed between the test instance and all training instances, then the instance k nearest neighbors are selected from the training data as per defined in the following figure.

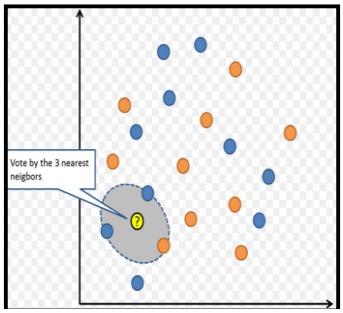

Fig. 2. 3-Nearest Neighbor

We choose SVM, because it is an immensely powerful classifier and it is more suited for 2-class problem. In addition, we compared experimentally SVM, Naïve Bayes and K-NN in performance and conclude that SVM has very good predictive power.

3.2 RapidMiner and Rapid Analytics:

The Community Edition of RapidMiner [2, 12] (formerly known as "Yale") is an open source toolkit for data mining. It provides the ability to easily define analytical steps and generate graphs. It is an environment for machine learning and data mining

experiments. RapidMiner provides a GUI which generates an XML (eXtensible Markup Language) file that defines the analytical processes the user wishes to apply to the data. This file is then read by RapidMiner to run the analyses automatically. While these are running, the GUI can also be used to interactively control and inspect running processes. RapidMiner can be used for text mining, multimedia mining, feature engineering, data stream mining and tracking drifting concepts, development of ensemble methods, and distributed data mining. RapidMiner provides data loading and transformation (ETL), data preprocessing and visualization, modeling, evaluation, and deployment. RapidMiner was rated as the fifth most used text mining software (6%) by Rexer's Annual Data Miner Survey in 2010. It is implemented in JAVA and available under GPL among other licenses. Internal XML representation ensures standardized interchange format of data mining experiments. GUI, command-line mode, and JAVA API allow invoking RapidMiner from other programs. In RapidMiner, several plugins are available for text processing, web mining etc. as well as a broad collection of data mining algorithms such as SVM, decision trees and self-organization maps.

Rapid Analytics [13] is the first open source business analytics server available. Rapid Analytics was built around the most widely used data mining solution RapidMiner and adds features like remote execution, scheduled processes, quick web service definitions, and a complete web-based report designer. Rapid Analytics is the new data mining server solution that uses RapidMiner both as a data mining engine and as a front-end to design data mining processes. We chose RapidMiner and Rapid Analytics for our implementation described in next section. First, it contains broad collection of plugins as well as large number of supervised learning methods. Second, classification engines created in RapidMiner but can be stored in remote repository to execute it remotely on the Rapid Analytic server at regular time interval.

4 Proposed Technique

In this section, we present our approach that includes (i) preparing a custom dictionary to label reviews as truthful or deceptive; (ii) the 'deceptive' classifier to predict testing data as a deceptive or truthful (iii) PHP script to label review as useful or not-useful; (iv) 'useful' classifier to predict testing data is either useful or not-useful; (v) "ranking" model to rank the reviews.

4.1 **Spam Review Detection:**

In general, spam review detection can be regarded as a classification problem with two classes, fake and non-fake. Machine learning models can be built to classify each review as deceptive or truthful. To build a classification model, we need labeled training examples of both classes. There was no labeled dataset for product opinion spam prior to this project. Recognizing whether a review is a deceptive opinion spam is extremely difficult if it has to be done manually reading the review because one can carefully craft a spam review which is just like any other genuine review. We prepared the dictionary for fake and non-fake reviews by adding knowledge from the dataset which is available on http://www.cs.uic.edu/~liub/FBS/CustomerReviewData.zip and using Cornell model. To prepare dictionary we passed reviews through Cornell model that tokenizes words based on specialized characters (like space, full stop, exclamation,

question mark etc.) in each sentence and puts it into any one of the appropriate category along with weight like high positive (+3), moderate positive (+2), low positive (+1), neutral (0), high negative (-3), moderate negative (-2) or low negative (-1). Some of words from neutral category of Cornell model are important for our domain and we placed those important words into positive or negative category with weight from http://www.cs.uic.edu/~liub/FBS/CustomerReviewData. After putting each word of each sentence into any one of six categories along with weight, we calculated final weight for each unique word based on our formula as follows:

$$Weight\ of\ each\ word = \frac{WordCount * Weight}{TotalWordCount}$$

More precisely we can say that,

$$Weight\ of\ each\ non\text{-}fake\ word = \frac{WC_{HP} * 3 + WC_{MP} * 2 + WC_{LP} * 1}{TotalWordCount_{Positive}}$$

where WC_{HP} is the count of a particular word in high positve category, WC_{MP} is the count of a particular word in medium positve category, WC_{LP} is the count of a particular word in low positve category.

$$Weight\ of\ each\ fake\ word = \frac{WC_{HN} * -3 + WC_{MN} * -2 + WC_{LN} * -1}{TotalWordCount_{Negative}}$$

where WC_{HN} is the count of a particular word in high negative category,

WC_{MN} is the count of a particular word in medium negative category

WC_{LN} is the count of a particular word in low negative category

\Using above formula, we prepared two wordlists for fake and non-fake reviews along with their corresponding weights. We called that dictionary through a php script to label the review as fake or non-fake based on final summation of all words in each review. If final summation of weight for fake and non-fake words of a review are positive then it is labeled as “non-fake” otherwise it is labeled as “fake”.

Building Models Using LibSVM

The first component of the framework is the ‘deception’ classifier, which predicts whether each unlabeled review is non-fake (truthful) or fake (deceptive). As mentioned previously, we labeled training review as deceptive or truthful, so that we can train ‘deception’ classifiers using a supervised learning algorithm. We tried three supervised learning algorithms: support vector machine (SVM), K-NN, Naive Bayes to classify product review using two pre-classified training sets: deceptive and truthful. Our work has shown that SVM trained and performs well in deception detection tasks. We found that SVM creates a hyper plane to best separate the two planes and it outperforms the other two classifiers. We trained SVM classifiers using software package of RapidMiner tool. Results of evaluation are presented in the next section.

4.2 Useful Review Detection:

In general, useful review detection can be regarded as a classification problem with two classes, useful and not-useful. Machine learning models can be built to classify each review as useful or not-useful. To build a classification model, we need labeled training examples of both useful and not-useful class. There was no labeled dataset for product opinions as useful and not-useful at the time of project (to the best of our knowledge). However, to recognize review is useful or not, we considered reader’s

rating on consumer's review. Using php we labeled reviews as useful if reader's rating is greater than 40% or as a not-useful review, if reader's rating is less than 40%.

Building Model Using LibSVM

The second component of the framework is 'useful' classifier, which predicts whether each unlabeled review is Useful or Not-Useful. As mentioned above, we labeled training data, so that we can train 'useful' classifiers using a supervised learning algorithm. We tried different supervised algorithms like Naïve Bayes, K-NN, and SVM. Our work has shown that SVM trained and performs well in useful or not-useful detection tasks as compared to other algorithms. We train SVM classifiers using the software package of RapidMiner tool. Results of the evaluation are presented in the next section.

4.3 Ranking Reviews:

The last component of the framework is the 'Ranking' Model. This model takes the output from the 'deceptive' classifier and 'useful' classifier as input to rearrange the reviews based on weight (confidence) of fake, non-fake, useful, and not-useful. Higher sort priority is given to deceptive/truthful reviews and then to useful/not-useful reviews. Results of evaluation of the 'ranking' model are presented in the next section.

4.4 Implementation:

For the implementation of our approach we used RapidMiner, XAMPP, Rapid Analytics tools. We created a PHP script to collect product (e.g. camera) reviews from amazon and Epinion sites. To label training data, we prepared dictionary of words for deceptive/truthful reviews and labeled the reviews by using the dictionary in the PHP script. We created another PHP script to label training set as useful or not-useful based on reader's rating. We utilized RapidMiner tool and its supervised learning method, e.g. SVM, for building the "deceptive" classification model and "useful" classification model as well as "ranking" model. For testing purpose, we designed HTML page to enter a product review. This review is stored in a database and when the RapidMiner process is executed, it will fetch reviews from the database and based on the classifier it is processed and results (reviews with classification) are stored in the database. Using the HTML page, the result of both classifiers can be displayed.

5 Experimental Results

For evaluation, we trained both our models using different types of datasets such as balanced and imbalanced. The training dataset for 'deceptive' classifier had 1348 reviews in the imbalanced dataset and 140 reviews in the balanced dataset. The training dataset for the 'useful' classifier had 5003 reviews in the balanced dataset and 5103 in the imbalanced dataset. The following experimental result show that 'deceptive' classifier gives better performance using imbalance dataset and 'useful' classifier performs well using balanced dataset with SVM classification algorithm. We calculated the performance of our models using the following formula.

Performance, $G = \sqrt{(Sn * Sp)}$

where Sn is the sensitivity and Sp is the specificity

$Sn = \frac{TP}{TP+FN}$ and $Sp = \frac{TN}{TN+FP}$

where TP is the number of true positives

TN is the number of true negatives

FP is the number of false positives

FN is the number of false negatives

Table 1: Fake/Non-Fake Classifier Performance

Algorithm	Performance (G=sqrt(spec*sens)) Imbalance Data - 1348	Performance (G=sqrt(spec*sens)) Balance Data - 140
LinearSVM	65.58%	70%
K-NN	62.18%	64.22%
Naive Bayes	60.34%	69.98%

We observed that SVM trained and performed well in deception detection tasks. We found that SVM creates a hyper plane to best separate the two planes and it outperforms the other two classifiers with an accuracy peak at about 66%. Cross-validated classifier performance results are presented in Table 1.

We tried different supervised algorithms like Naïve Bayes, K-NN, and SVM for "Useful" classifier. Evaluation results show that SVM trained and performed well in useful or not-useful detection tasks as compared to other algorithms. This approach has been evaluated to be nearly 78% accurate at detecting useful or not-useful in a balanced dataset. Cross-validated classifier performance results are presented in Table 2. Results of the ranking model are presented in Table 3.

Table 2: Useful/Not Useful Classifier Performance

Algorithm	Performance (G=sqrt(spec*sens)) Balance Data (5003)	Performance (G=sqrt(spec*sens)) Imbalance Data (5103)
LibSVM	78.29%	70%
K-NN	77.07%	72.58%
Naive Bayes	73.79%	73.05%

Table 3: Top Ranked Reviews

	Truthful/Deceptive		Useful/Not-Useful	
Consumer Reviews For Product	Result	Confidence	Result	Confidence
bought thi camera becau light photo capabl disappoint need camera could n	deceptive	87.60%	Useful	52.60%
first love canon camera have gotten point where will onli canon camera tri	truthful	84.70%	Useful	58.00%
thi excel camera class len particular impress macro telephoto light would be	truthful	83.90%	Not Useful	52.80%
soni cybershot went dead look someth replac someth afford price compact	truthful	83.30%	Useful	84.70%
thi camera near splurg purcha part regret find that grab thi camera when le	truthful	82.30%	Useful	78.70%
bought thi camera special snorkel littl scuba photo camera great just snorke	deceptive	76.80%	Not Useful	51.30%
bought thi upgrad canon thank their menu featur similar like learn whole cai	deceptive	75.50%	Useful	69.80%
befor went bigger zoom great camera great pictur amaz video super zoom	deceptive	74.30%	Not Useful	62.60%
prior canon broke dai befor christma rush store plai with canon fell love wit	deceptive	73.50%	Useful	51.90%
have just bought thi canon powershot current stai australia price australia hi	deceptive	73.30%	Not Useful	53.40%
bought thi canon befor trip europ happi that qualiti pictur veri good even wh	deceptive	73.30%	Useful	55.70%

6 Summary and Future Work

As individuals and businesses are increasingly using reviews for their decision-making, it is critical to detect spam reviews. We presented our approach for detecting spam, not-useful reviews and prioritization of the reviews based on their weight (confidence). The evaluation shows that 'deceptive' classifier and 'useful' classifier is nearly 66% and 78% accurate respectively. Various supervised learning methods were used and we observed that SVM worked best as it is an immensely powerful classifier and it is well suited for 2-class problem. In addition, we compared experimentally SVM, Naïve Bayes and K-NN in performance and concluded that SVM has very good predictive power. Online reviews are worthless if they are not honest opinion. Our models, can give an idea to users on which reviews are non-fake and useful as well as which reviews should be completely ignored in product purchase decision-making thereby helping choose the right product. Future work might explore other methods for labeling online reviews, and will focus on improving the accuracy and more sopphisticated techniques for detecting spam reviews.

References

[1] S. B. Kotsiantis, I. Zaharakis, P. Pintelas. "Supervised machine learning: A review of classification techniques." Emerging artificial intelligence applications in computer engineering 160 (2007): 3-24.

[2] M. Hofmann, R. Klinkenberg, eds. RapidMiner: Data mining use cases and business analytics applications. CRC Press, 2013.

[3] C. Danescu-Niculescu-Mizil, G. Kossinets, J. Kleinberg, and L. Lee. How opinions are received by online communities: a case study on amazon.com helpfulness votes. In 18th international conference on World Wide Web (WWW), pp. 141-150, 2009.

[4] C. Dellarocas. Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In ACM Conference on Electronic Commerce (EC), pp. 150-157, 2000.

[5] I. Fette, N. Sadeh-Koniecpol, A. Tomasic. Learning to Detect Phishing Emails. In Proceedings of International Conference on World Wide Web (WWW), 2007.

[6] N. Jindal and B. Liu. Opinion spam and analysis. In WSDM, 2008.

[7] S.-M. Kim, P. Pantel, T. Chklovski, M. Pennacchiotti. Automatically assessing review helpfulness. In Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 423-430, 2006.

[8] N. Jindal and B. Liu. Analyzing and Detecting Review Spam. In IEEE Intl. Conference on Data Mining (ICDM), pp. 547-552, 2007.

[9] J. Liu, Y. Cao, C.-Y. Lin, Y. Huang, and M. Zhou. Low-quality product review detection in opinion summarization. In Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), 2007.

[10] G. Wu, D. Greene, B. Smyth, and P. Cunningham. Distortion as a validation criterion in the identification of suspicious reviews. In Proceedings of the First Workshop on Social Media Analytics (SOMA) at SIGKDD, pp. 10-13 , 2010.

[11] M. Ott, C. Cardie, and J. Hancock. Estimating the prevalence of deception in online review communities. In Proceedings of the 21st international conference on World Wide Web (WWW), pp. 201-210, 2012.

[12] RapidMiner: http://www.softwarenhardware.com/tag/rapidminer-tutorial/ [Last Accessed: Mar 2018]

[13] Rapid Miner & Rapid Analytics [Online] http:// www.rapidi.com/downloads/brochures/RapidMiner_ Fact_Sheet.pdf [Last Accessed: March 2018]

[14] A-M. Popescu, O. Etzioni. Extracting Product Features and Opinions from Reviews. EMNLP'2005.

[15] T. Seidl, H. Kriegel. "Optimal multi-step k-nearest neighbor search." ACM Sigmod Record. Vol. 27. No. 2. ACM, 1998.

[16] L. Li, B. Qin, W. Ren, T> Liu. "Document representation and feature combination for deceptive spam review detection." Neurocomputing, Volume 254, 2017, pp. 33-41.