Noname manuscript No.
(will be inserted by the editor)

Words, Tweets, and Reviews: Leveraging Affective Knowledge Between Multiple Domains

Felipe Bravo-Marquez · Cristián Tamblay

Received: date / Accepted: date

Abstract Background Three popular application domains of sentiment and emotion analysis are: 1) the automatic rating of movie reviews, 2) extracting opinions and emotions on Twitter, and 3) inferring sentiment and emotion associations of words. The textual elements of these domains differ in their length i.e., movie reviews are usually longer than tweets and words are obviously shorter than tweets, but they also share the property that they can be plausibly annotated according to the same affective categories (e.g., positive, negative, anger, joy). Moreover, state-of-the-art models for these domains are all based on the approach of training supervised machine learning models on manually-annotated examples. This approach suffers from an important bottleneck: manually annotated examples are expensive and time-consuming to obtain and not always available.
Methods In this paper we propose a method for transferring affective knowledge between words, tweets, and movie reviews using two representation techniques: Word2Vec static embeddings and BERT contextualized embeddings. We build compatible representations for movie reviews, tweets, and words, using these techniques and train and evaluate supervised models on all combinations of source and target domains.
Results and Conclusions Our experimental results show that affective knowledge can be successfully transferred between our three domains, that contextualized embeddings tend to outperform their static counterparts, and that better transfer learning results are obtained when the source domain has longer textual units that the target domain.

Keywords Transfer learning · sentiment analysis, affect in language

Felipe Bravo-Marquez (Corresponding author)
Department of Computer Science, University of Chile
Millennium Institute for Foundational Research on Data, IMFD-Chile
E-mail: fbravo@dcc.uchile.cl

Cristián Tamblay
Department of Computer Science, University of Chile
E-mail: ctamblay@dcc.uchile.cl

1 Introduction

The field of sentiment analysis attempts to computationally extract people's opinions, emotions and views from natural language texts. A closely related field is affective computing, which focuses on the design of machines capable of recognizing and expressing human emotions [8]. Finally, the field of sentic computing proposes a holistic view of human emotions and natural language that integrates both sentiment analysis and affective computing with other related disciplines such as knowledge representation, linguistics, and psychology [9].

These techniques have been successfully applied in various domains, such as automatic monitoring of public opinion in social media, conducting market research and improving companies' customer service. As a more concrete example, in [21], the authors applied sentiment analysis techniques to tweets related to the 2016 U.S. presidential election, and reported a correlation of 94% with official polls. Authors in [49], predicted movie ratings on RottenTomatoes[1], a website where experts assign ratings to movies, using movie reviews as input data.

A particular property of sentiment and emotions is that they can be found across all types of linguistic units (e.g., words, phrases, sentences, paragraphs, documents) and textual sources (e.g., social media publications, movie reviews, newspapers). In this paper, we study how the affective[2] knowledge between three different domains can be leveraged and ultimately transferred: words, tweets, and movie reviews.

We argue that because of the semantic interaction between words, sentences, and documents, the affective patterns between these three domains are strongly interconnected, something that has been widely studied by the linguistics community as discussed below.

The principle of semantic compositionality claims that the meaning of a sentence is a function only of the meaning of its lexical units, together with how these units are combined [38]. This principle suggests that the meaning of a sentence is determined by the meaning of its individual words as well as the sentence structure. The distributional hypothesis, on the other hand, states that words used in the same contexts tend to have similar meanings [19]. As a consequence, word meanings can be inferred by the contexts in which they occur.

These two linguistic theories propose a conceptual framework of meaning for both words and sentences, which we extend in this work to affective states such as sentiment and emotions. The relationship between meaning and affect can be argued at both the lexical and sentence level. At the lexical level it can be argued that words with similar meanings (i.e., synonyms) probably express the same sentiment, and similarly, sentences that convey the same meaning using alternative expressions are also very likely to express the same sentiment and the same emotions.

These principles are used in this research to find exploitable patterns in the relationship between words, sentences, and documents. More specifically, we focus on specific sentiment and emotion detection tasks described below.

[1] www.rottentomatoes.com

[2] We use the term "affect" to encompass both sentiment and emotions.

The sentiment analysis tasks that we study are: 1) polarity lexicon induction (PLI) [2], 2) sentence-level sentiment classification (SSC) [35], 3) and document-level sentiment classification (DSC) [3]. The objective of the PLI task is to determine the semantic orientation of a word in a lexicon, which corresponds to classifying whether the word is positive or negative. For example, it classifies the word "happy" as positive and the word "sad" as negative. Meanwhile, the SSC task aims to classify entire sentences as positive or negative. An example of this task would be to classify the tweet "my dog is the best #doglover" as positive[3]. Finally, the DSC task intends to classify entire documents as positive or negative, based on the opinion expressed in them. Movie reviews are a clear example of this task.

We also study two emotion tasks at both word and sentence level[4]. The first task is the detection of word affect intensities (WAI) [34], which consists of associating words with real-valued intensity scores for four basic emotions: anger, fear, sadness, and joy. For example, the word "outraged" has a higher intensity for the emotion anger than the word "agitated". The second task is the detection of sentence-level affect intensities (SAI), which consists of detecting the intensity of emotion felt by the speaker of a tweet [32].

All the above tasks share the property of being addressed by supervised learning algorithms trained on numerical vector representations of their corresponding lexical units and manually annotated labels. However, manually annotating words, tweets, and movie reviews into affective categories can be very time consuming and expensive.

In many practical scenarios, the resources needed for training supervised models (i.e., annotated examples) are not available. A possible solution to this problem, is to adapt models trained from a related domain where training data is available, to the task at hand.

In the context of sentiment classification, training a model in the word domain and then applying it in a sentence classification task (or vice versa) has been shown to be useful when training data from the target domain is insufficient [6]. This exercise is commonly known as transfer learning. Transfer learning between two domains refers to the acquisition of knowledge from a source domain and its subsequent application to a target domain. A formal definition of transfer learning is given as follows:

Definition 1 (Transfer learning) Given a source domain $\mathscr{D}_S$ and a learning task $\mathscr{T}_S$, a target domain $\mathscr{D}_T$ and a learning task $\mathscr{T}_T$, transfer learning aims to help improve the learning of the target predictive function f_T in $\mathscr{D}_T$ using the knowledge $\mathscr{D}_S$ and $\mathscr{T}_S$, where $\mathscr{D}_S \neq \mathscr{D}_T$, or $\mathscr{T}_S \neq \mathscr{T}_T$ [37]

Transfer learning requires both the source and target domains to be related. The three domains we focus in this study (words, tweets, and movie reviews) are related in the sense that they can all be plausibly associated with the same affective categories (e.g, positive, negative, anger, joy).

Each one of our domains expresses affect in a unique way. First, tweets capture someone's mood and thoughts in a short but emotionally charged message. Then,

[3] In this work we make the assumption that tweets are usually formed by a single sentence.

[4] We do not study emotions at the document level due to the lack of annotated data to experiment with.

movie reviews comprehensively reflect why someone liked or disliked a movie. Finally, the field of lexical semantics has widely studied the inherent affect of isolated words, which is also referred to as semantic orientation [15]. Another important difference between our domains is the length of their respective linguistic units: movie reviews are typically longer than tweets, and tweets are composed by words.

The core of our transfer learning proposal is to represent the lexical units of each domain with compatible representations (i.e., numerical vectors residing in the same vector space). Afterwards, a classifier can be trained on labeled instances from a source domain, to be later deployed on instances from a target domain. Any of our three domains (i.e., words, tweets, and movie reviews) can interchangeably play the role of source or target domain.

The most important building block that our approach requires, is a model capable of representing lexical units of different lengths as compatible feature vectors. In this work, we adapt two popular resources in Natural Language Processing (NLP) for this purpose: 1) static word embeddings [30], and 2) contextualized word embeddings [14].

The main difference between them is that while static word embeddings assign a fixed representation to each word, contextualized word embeddings provide a variable representation that depends on both the word and its context.

Specifically, we experiment with all the affective tasks and combinations of source and target domains described above, using the following representation models:

- BERT-Base contextualized word embeddings [14].
- Word2Vec [30] static word embeddings trained on the same dataset as BERT-Base (i.e., Wikipedia + BookCorpus [60]).
- Word2Vec [32] static word embeddings trained over a corpus of tweets.

The main contribution of this work is a new framework that allows a transparent comparison of static and contextual word embeddings for various scenarios of affective knowledge transfer between words, tweets, and movie reviews. We argue that this approach can be especially valuable when annotated data in the target domain is scarce. Moreover, we also believe that our proposal can benefit the sentic computing community, as it allows for directly transfering affective knowledge from lexical resources such as SenticNet [10] to other domains.

The remainder of this article is organized as follows. Section 2 presents a background in NLP and language representation relevant to this article. In Section 3 we provide a review of related work in affect analysis. In Section 4, we describe the proposed method for transferring affective knowledge. In Section 5, we present the experiments conducted to evaluate transfer learning tasks and discuss results. The main findings and conclusions are presented in Section 6.

2 Background

In this section we present several concepts of Natural Language Processing (NLP) that are used in this article. The goal of NLP is to enable computers to process and analyze human language. Lexical semantics, which is a particular sub-area of NLP, aims to find hidden semantic relationships between lexical units such as words.

The simplest way to turn a word into a vector is to use the one-hot encoding model, that is, a sparse vector of the size of the vocabulary, containing a value of 1 in the corresponding word's index. This model can naturally be extended to represent a document by adding or averaging the one-hot encoded vectors of their words, which is referred to as the the bag of words model [19].

Due to the proliferation of deep neural networks in NLP, word and sentence representation models have experienced a dramatic evolution during the last decade that resulted in more powerful word and sentence representation models such static and contextualized "word embeddings" to be presented next.

On the one hand, static word embeddings define an injective mapping f between words and their respective d-dimensional representations, i.e.,

$$f : w_i \rightarrow \mathscr{R}^d \tag{1}$$

This means that one word has only one representation. On the other hand, contextualized word embeddings define an injective mapping f from a sequence of words of length n to a sequence of d-dimensional representations, i.e.,

$$f : \{w_1, w_2, ..., w_n\} \rightarrow (\mathscr{R}^d)^n \tag{2}$$

In this paradigm, each word has as many representations as contexts.

A salient property of these two models is their ability to learn from massive amounts of unlabeled corpora, which can be freely obtained from the Web.

Word2Vec [30] and GloVe [39] are possibly the most popular static word embedding models that can be efficiently trained on large corpora. These models can effectively capture semantic and syntactic properties of words by exploiting their surrounding words in a fixed sized window [7]. In [58], authors employed these models for sentiment classification, yielding results with an accuracy above 85%. Static word embeddings have improved the performance of a wide range of NLP tasks, such as machine translation [61] and text classification [22]. The main limitation of static word embeddings is that they conflate all the different meanings of polysemous words into a single representation.

Over the last years, novel contextualizers such as ELMo, BERT, [14] and XLNet [56] have dramatically improved performance for many (NLP) tasks, including sentiment analysis. ELMo and its predecessors extract context-sensitive features from left-to-right and right-to-left text representation models. This model advanced the state-of-the-art for several NLP benchmarks [40]. With the release of the Transformer architecture [52], LSTM-based neural networks, such as ELMo, started to fall behind mainly because Transformers can more efficiently deal with long-term dependencies [51]. BERT and XLNet are examples of deep learning architectures that use the Transformer architecture [52]. These architectures can concurrently process all inputs of a sequence, leading to faster training times. Because of this, novel Transformer-based architectures quickly gained more attention than LSTM-based ones. They can exploit the advantages of the new graphics processing units (GPUs) and be trained over massive datasets. The standard way to use these models consists of a pre-training phase, in which the model is built in a self-supervision scheme over a large corpus, and then a fine-tuning phase in which the model is adapted to the target task where

labeled examples are available. Alternatively, a pre-trained contextualizer can be used as a feature extractor without the need for fine-tuning [43].

As discussed so far, the idea of pre-training a neural network model from a large corpus to obtain language representations is now a standard practice in NLP. Below we summarize the most relevant architectures that follow this approach:

- **Word2Vec** [30]: this is a two-layer neural network that is trained over a corpus of documents and returns a set of feature vectors for the words in that corpus. Word2Vec model is used for learning vector representations of words. This model is based on the static word embeddings paradigm.
- **Recurrent Neural Network** (RNN): in these networks, a transformation is repeatedly applied to a sequence of input vectors to produce another sequence of output vectors. For example, the long short-term memory network (LSTM) is an RNN architecture in which the model is continuously fed with new inputs along with the previous state vector and can decide whether to add to or remove information from the state cell. LSTMs can keep track of arbitrary long-term dependencies relatively well. LSTMs serve as the basic building block of first generation contextualized models such as ELMo [40].
- **Attention**: an attention function is a map f from a query and multiple (key,value) pairs, i.e.,

$$f : \text{query} \times (\text{keys,values})^n \rightarrow \mathscr{R}^d \tag{3}$$

  The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a function [52]

$$w : \text{query} \times \text{key} \rightarrow \text{weight} \tag{4}$$

- **Encoder - Decoder**: An encoder is a stack of several recurrent units or attention mechanisms where each one accepts a single element of the input sequence, collects information for that element, and propagates it forward. A decoder takes the encoder vectors as input; it goes through several other recurrent units or attention mechanisms and produces an output [54].
- **Transformer**: this neural network relies entirely on an attention mechanism to capture global dependencies between input and output. The Transformer follows encoder-decoder overall architecture using stacked self-attention and point-wise, fully connected layers for both the encoder and decoder. The Transformer allows for significantly more parallelization than RNNs [52].

The success of BERT led to the development of many other models based on it, many of which have reported state-of-the-art results in various NLP tasks. Examples of these are DistilBERT [46], RoBERTa [24], ERNIE [59], BETO [11] and ELECTRA [13]. This success is the motivation for our work, and to the best of our knowledge, there are no studies on using BERT to transfer affective knowledge across multiple domains.

3 Related Work

In this section, we review works on transfer learning, sentiment analysis and sentic computing that are relevant to this article.

As explained in the introduction, transfer learning refers to the acquisition of knowledge from one source domain and its application to a related target domain. Prior work on transfer learning focuses on domain adaptation, i.e., training a classifier in one domain, e.g., internet blogs, and deploying it in a domain where a different terminology is used, e.g., newspapers. [17].

There is a proposed transfer learning framework for transferring sentiment knowledge between words and tweets based on the aggregation of instances in [6]. This work provides the foundations of our own work, representing tweets and words using compatible representations. Words are represented as the collection of tweets in which they occur and sentences are the centroid of each word representation. However, this paper relies on high-dimensional sparse representations such as word unigrams. In contrast, our work focuses on modern deep learning architectures and dense representations. Moreover, we also go beyond sentiment, analyzing also the intensities of emotions.

WordNet [31] is an English database created in the 90s that links nouns, verbs, adjectives, and adverbs to form sets of synonyms called synsets. These synonyms are, in turn, connected by semantic relationships that determine the definitions of words. This resource is particularly useful for handling polysemic words, that is, words with multiple meanings. In [15], a polarity classifier was trained using the WordNet database on a set of positive and negative labeled words. For each unknown word, related terms in which the polarity is known are retrieved (e.g., synonyms, antonyms) and used to classify the unknown polarities by assuming that synonyms must have the same polarity and antonyms the opposite. The resulting expanded lexicon is employed to determine the polarity of sentences by adding up the polarity of their words. This process can be considered as an example of transfer learning from the domain of words to the domain of sentences.

In [20], the authors adopted an approach where sentiment annotated Amazon reviews help transfer knowledge to the aspect-level sentiment classification task using an LSTM neural network architecture.

An algorithm based on the joint regularization of a bipartite graph of labeled and unlabeled nodes was proposed in [47]. The nodes correspond to documents and words, and sentiment labels are propagated from the labeled nodes to the unlabeled ones using regularized least squares.

In [29], a framework for incorporating word knowledge to text sentiment classification using a generative Naïve Bayes model was proposed. In that paper, the authors refine the knowledge of a sentiment lexicon with annotated blogs. They then proceed to classify movie reviews, political blogs, and IBM products reviews.

A recursive neural tensor network capable of learning the sentiment of lexical units of different granularities such as words, phrases (including negated expressions), and sentences was proposed in [48]. An unsupervised learning approach for Twitter sentiment analysis using three domain-independent sentiment lexical re-

sources [12], shows that lexicons can infer the sentiment of tweets by averaging the sentiment values of their lexical units.

In the following paragraphs, we present other works that fall within the sentic computing paradigm and are relevant to this study.

A relevant problem in sentiment analysis is contextual polarity ambiguity. This problem is tackled in [55] by performing word polarity disambiguation using Bayesian model with opinion-level features. In [25], the authors proposed a novel neural network architecture and two extensions of the traditional LSTM architecture for the task of targeted aspect-based sentiment analysis, i.e., recognizing aspect categories and assigning their polarity. The work of [42] introduces a novel paradigm for concept-level sentiment analysis that combines the Hourglass of Emotions [50], common-sense computing, and deep learning techniques. In [1], authors propose a stacked ensemble method for predicting real-valued intensities of emotion and sentiment. The method combines the outputs of various deep learning and standard feature-based models using a feedforward network. Finally, [44] applied a convolutional neural network for sentiment classification of Hindi movie reviews that outperformed many other machine learning baselines.

4 Proposed Methodology

This section starts by introducing our transfer learning method, followed by a description of how to represent words and sentences using static word embeddings and ends with another description of how to achieve the same goal using contextualized word embeddings.

4.1 Method

This subsection introduces a new method for transferring affective knowledge between words, sentences, and movie reviews based on two paradigms of language representation. As stated in Section 1, the principle of semantic compositionality states that the meaning of a sentence depends on its lexical elements together with the form in which they are composed. On the other hand, the distributional hypothesis suggests that word meanings can be inferred by the contexts in which words occur.

These two semantic theories lay the foundation of our study, giving us the tools to jump from the word domain to the sentence or document domains and vice versa. Our method is inspired by previous work to transfer sentiment knowledge word and tweets [6]. The method is illustrated in Figure 1. One domain will act as the source domain $\mathscr{D}_S$ and the other as the target domain $\mathscr{D}_T$. The transfer learning procedure is described in the following steps:

1. Set the language representation model, choosing between static or contextualized word embeddings.
2. Represent both the training instances from the source domain $\mathscr{D}_S$ and the testing instances from the target domain $\mathscr{D}_T$ with the chosen representation model to

obtain compatible k-dimensional vectors, using aggregation functions explained in the next subsections.
3. Train a predictive model on the training instances represented according to previous step.
4. Apply the resulting model to the corresponding testing instances from $\mathscr{D}_T$.

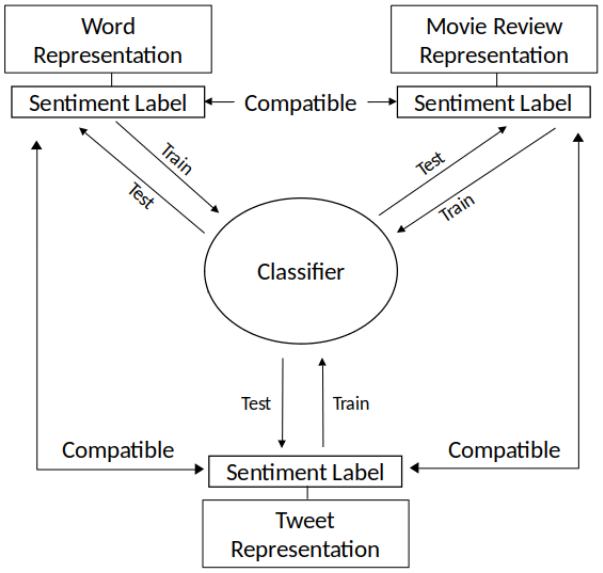


Fig. 1: Method for transferring sentiment knowledge between words, tweets and movie reviews

As an example, let us assume a scenario of transferring sentiment knowledge between words to tweets. We have the following annotated words {"hate": negative, "love": positive} and the following annotated tweets {"I detest this movie": negative, "I love you": positive}. First, we represent both datasets with the chosen paradigm, obtaining compatible representations. Then, we train a classifier on the word's dataset and predict the sentiment of the target tweets, using the corresponding representations in both cases. Finally, we evaluate the overall performance of the classifier on the target domain. We expect that after obtaining sentiment knowledge from the word domain, the classifier will be able to successfully classify the sentiment of tweets as a result of the semantic relationships between these two domains. The same could be done in the other direction, training on the tweet domain and testing on the word domain. This process is illustrated in Figure 2.

It is worth pointing out that the classifier will not be further adjusted (i.e., by tuning hyper-parameters) using data from the target domain, since we are interested in evaluating the model's transfer learning capabilities in a scenario where there is no training data in the target domain.

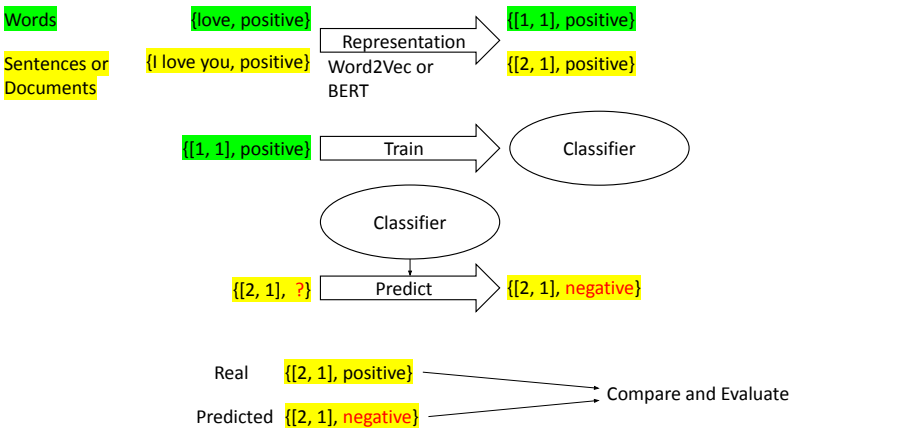


Fig. 2: Illustrative example of transferring sentiment from words to tweets using 2-dimensional representations.

Having introduced the fundamental ideas of our methodology, we now describe how words, sentences, and documents are represented using our two aforementioned paradigms.

4.2 Static word embeddings

As discussed in Section 2, static word embeddings define an injective mapping between words and vectors:

$$f : w_i \rightarrow \mathscr{R}^d. \tag{5}$$

This means that a word has one and only one representation. These static word embeddings are obtained by training neural networks on large corpora. There are some hyper-parameters that need to be adjusted such as the context's size and the embeddings vectors' size. The context size is usually known as the window size and refers to the number words surrounding each target word that are considered during the learning phase. The embeddings vectors' size is usually set between 100 to 300 dimensions. Very low dimensional embeddings are usually incapable of capturing rich semantic information. On the other hand, there is no evidence of significant gains by increasing the dimessionality after a certain point.

Static word embeddings need an aggregation function to pass from word embeddings to sentence embeddings. Usually, this aggregation function is a linear map over the individual word embeddings in a sentence. We will use the "average" aggregation function. Out-of-vocabulary words do not have a representation in some static word embeddings models, so they are removed. Formally, given a collection $\{w_1,...,w_n\} \in \Sigma^+$ of words in a sentence where Σ are the symbols of the language. First we remove out-of-vocabulary words, resulting in a collection $\{w_1,...,w_k\}$. Then

we obtain the embedding as follows:

$$Embedding(\{w_1,...,w_n\}) = \frac{1}{k}\sum_{i=1}^{k} Embedding(w_i) \tag{6}$$

If $\frac{k}{n} < 0.8$, i.e., more than 20% of tokens are unknown, the sentence is discarded. Words do not need an aggregation function as they are represented passing them through the model.

The process by which words and sentences are represented with compatible vectors is illustrated in Figure 3. Words and sentences are represented by two-dimensional integer vectors for the sake of simplicity.

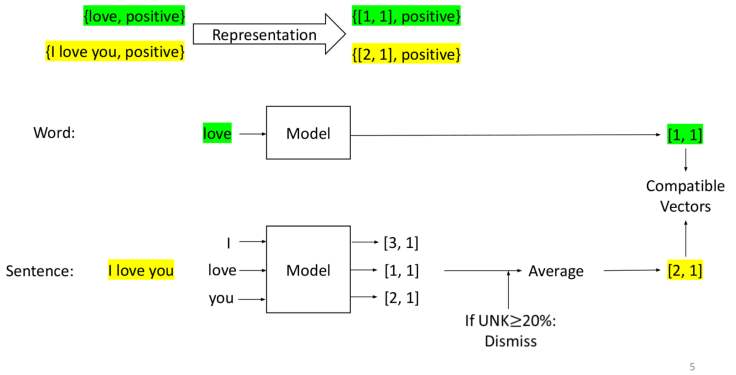


Fig. 3: Illustrative example representing words an sentences with compatible 2-dimensional vectors using Word2Vec.

4.3 Contextualized word embeddings

As shown in Section 2, contextualized word embeddings define an injective mapping between a sequence of tokens to a sequence of vectors:

$$f : \{w_1, w_2, ..., w_n\} \rightarrow (\mathscr{R}^d)^n. \tag{7}$$

In this paradigm, each word would have a different representation depending on the context.

More specifically, contextualizers receive a sequence of known tokens as input. These tokens are obtained by a special tokenizer, which maps words to IDs in the model's dictionary. As an example, BERT uses WordPiece tokenizer in which most

frequent combinations of the symbols in the vocabulary are iteratively added to the vocabulary. This is useful for treating rare and out of vocabulary words, e.g., splitting the word *strawberry* into *straw* and *#berry* that are more common.
By the nature of contextualizers, sentences do not need any special treatment. They are passed directly through the tokenizer and then fed into the model. It is essential to add that BERT supports up to 512 tokens, so any sequence above that is truncated. In the case of words, we consider them a single word sentence and then use the same procedure. Impressively, this way to represent words yields good results.

There are many ways to obtain words or sentence representations using contextualizers. The most common approach is to average or sum the model's last hidden layer. In BERT, we can also use the output of the first token, known as the CLS token, which is usually employed for text classification tasks. These aggregation methods are used to represent both sentences and word because words can also have multiple tokens under BERT's tokenization scheme.

The process by which words and phrases are represented with compatible vectors using BERT is illustrated in Figure 4. We assume that the last hidden layer of BERT is a two-dimensional integer vector. Note that compatible vectors can be obtained either from the CLS token representation or by averaging the vectors of the other tokens. The last token [SEP] is discarded from the final representation.

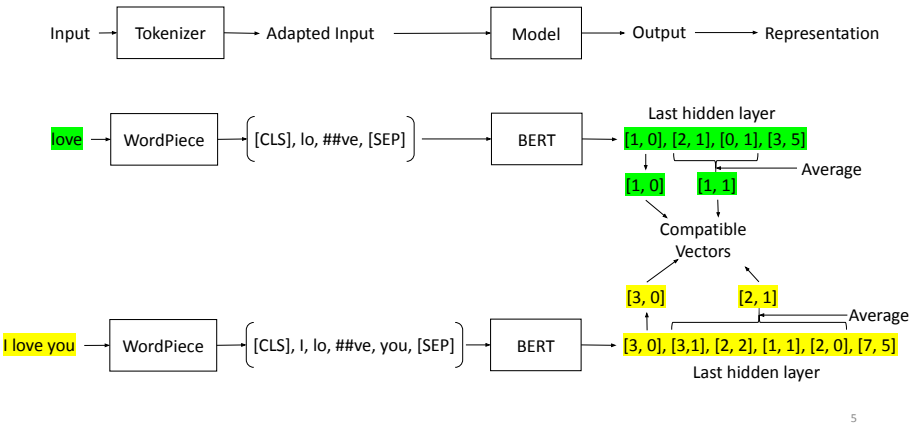


Fig. 4: Illustrative example representing words an sentences with compatible 2-dimensional vectors using BERT.

5 Experiments

In this section, we report transfer learning experimental results. We divide it into five subsections. The first subsection describes the datasets and lexicons used. Then, we present the representation models used in Subsection 5.2. Next, Subsections 5.3 and 5.4 report sentiment experiments and discussions. In Subsection 5.5 and 5.6 we

describe and discuss our emotion experiments. Finally, in Subsection 5.7 we conduct a qualitative analysis of prediction errors.

5.1 Data

Our sentiment experiments require three annotated resources for training and evaluation purposes, each of which coming from one of our domains: documents, sentences (or tweets), and words.

The document domain dataset is a collection of $50,000$ English reviews from the Internet Movie Database (IMDB) dataset [26]. This dataset consists of an equal number of positive and negative reviews, considering only the highly polarized ones. We use the official split of this dataset, that is, 50% training and 50% test.

The second dataset is the SemEval-2014 Task 9 [45] corpus, consisting of $5,232$ positive and $2,067$ negative tweets annotated by Amazon Mechanical Turk. We split the dataset stratified and randomly into 50% training and 50% testing for evaluating the transfer learning tasks.

We consider the *metaLex* dataset [5] as the word domain lexicon. This resource is built from the combination of four existing lexicons: *MPQA* [53], *Bing Liu* [23], *Afinn* [36], and *NRC-emotion lexicon* [33] resulting in $17,271$ positive, neutral and negative words. We kept only the positive and negative words and discarded those with conflicting polarities according to different lexicons. This results in a collection of $10,183$ annotated words. We split this lexicon using random and stratified partitions of 67% training and 33% testing instances. The main properties of the three datasets are summarized in Table 1.

Dataset	metaLex	IMDB	SemEval 2014
Positive Instances (Train)	2,525	12,500	2,642
Negative Instances (Train)	4,295	12,500	1,007
Positive Instances (Test)	1,244	12,500	2,590
Negative Instances (Test)	2,116	12,500	1,060

Table 1: Sentiment datasets properties.

We also evaluate the task of transferring four emotion intensities (anger, fear, sadness, and joy) between words and tweets. The annotated tweets are taken from the WASSA-2017 Shared Task on Emotion Intensity [32]. This dataset is composed of $7,097$ tweets and is divided into 4 separate datasets, each for a different emotion: *anger*, *fear*, *sadness*, and *joy*. Each tweet receives a real-valued score that determines the strength of the corresponding emotion in a range between 0 and 1. These tweets were annotated using the best-worst scaling technique. We use the official training and testing partitions of this dataset as shown in Table 2.

Our word-level emotion labels are obtained from the NRC Affect Intensity Lexicon v0.5 (NRC-AIL) [34]. This lexicon contains intensity scores for four basic emotions: *anger*, *fear*, *sadness*, and *joy*, rated between 0 and 1, as in WASSA, making both datasets compatible. This lexicon was also built using the best-worst scaling

technique, and each of its 4,192 words may be associated with multiple emotions. To deal with this, we created independent training and testing partitions for each emotion with a 67-33 train-test split. Table 2 features NRC lexicon partitions.

Dataset	NRC	WASSA
Anger Instances (Train)	994	941
Anger Instances (Test)	489	760
Fear Instances (Train)	1,183	1,257
Fear Instances (Test)	582	995
Sadness Instances (Train)	870	860
Sadness Instances (Test)	429	673
Joy Instances (Train)	849	897
Joy Instances (Test)	418	714

Table 2: Emotion datasets properties. Words in NRC lexicon may be associated with multiple emotions.

5.2 Representation models

As discussed in Section 4, we experiment with two approaches to representing words and sentences: static and contextualized word embeddings.

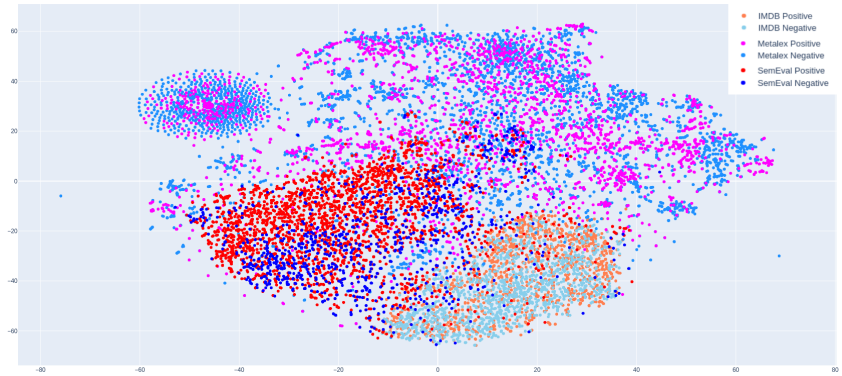
Regarding the contextualized word embeddings we use the BERT-base model in our experiments. This model has 12 layers (transformer blocks), 12 attention heads, and 110 million parameters in total. It was trained on the union of two text corpora: The English Wikipedia and BookCorpus [60],

Regarding the static word embeddings, we use two Word2Vec [30] models, which we refer to as *General purpose embeddings* and *Edinburgh embeddings*. We trained General purpose embeddings on the same data as BERT-base (English Wikipedia + BookCorpus), setting the embedding dimension to 300 and the window size to 15. The rationale for using these corpora is to be able to compare the results of Word2Vec and BERT when both resources were built on the same data. We also use Edinburgh embeddings, a Word2Vec model trained over the Edinburgh dataset [41] consisting of 10 million tweets. The hyper-parameters of this model were calibrated on an emotion classification task [4] and correspond to a window size of 5 and 400 dimensions.

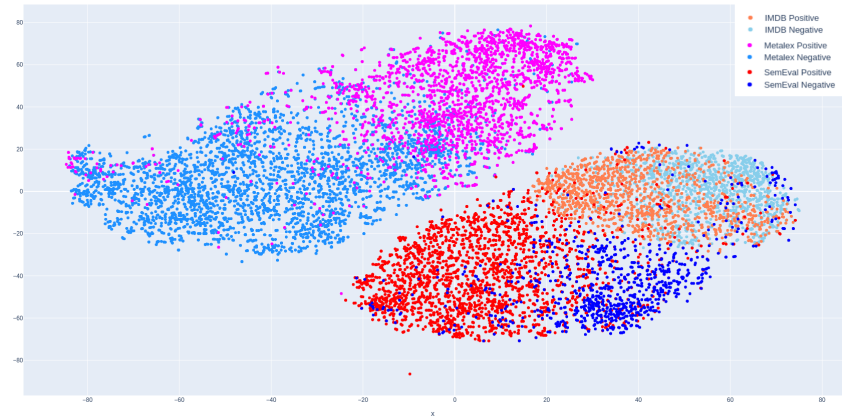
In Figure 5 we show a 2-dimensional visualization of the three sentiment training datasets (IMDB, Metalex and SemEval) using the t-SNE dimensionality reduction technique [27] for both Word2Vec Edinburgh embeddings and BERT average representation.

Before analyzing the figure, it is important to note that t-SNE is an unsupervised technique and therefore the 2-dimensional projections are computed without taking the labels into account. Despite this, we can see that, particularly in the case of BERT, negative and positive examples tend to lie in clearly separated regions. This suggests that BERT implicitly captures sentiment better than Word2Vec. It is also worth noting that words are distributed differently in the latent space than tweets and movie

reviews. This pattern is very clear for the case of BERT, where words do not overlap with examples from the other domains. We attribute this to the fact that words are less affected by the averaging operator, which has the effect of canceling out extreme values in the latent space. We will see in the next section that this is not a limitation for our method, since the decision boundaries learned by our classifiers are in many cases able to successfully separate positive and negative examples for the other domains.



Word2Vec Edinburgh



BERT AVG

Fig. 5: 2D visualization of all sentiment training datasets projected to the same space using t-SNE for both BERT AVG and Word2Vec Edinburgh representations. Point colors indicate the sentiment/dataset combination associated with each example.

5.3 Sentiment Experiments

In this subsection, we report the results of sentiment classification experiments using static and contextualized word embeddings. We study the effect of training a logistic regression classifier on different domains and then testing on words, tweets, and movie reviews, all of them labeled by positive or negative sentiment.

We use Weka[5] [18] for classification tasks, in particular the L2-regularized logistic regression method implemented in the LibLINEAR [16] package.

We use three metrics to measure the performance of our binary classification models: ROC AUC score, macro-averaged F1 score, and Cohen's Kappa score. Table 3 shows the results of the embeddings trained over BookCorpus and the English Wikipedia, and Table 4 shows the results using Edinburgh embeddings.

		Test domain			
		Word	Reviews	Tweets	
Train domain	Word	0.949	0.784	0.769	AUC
		0.881	0.603	0.684	F1
		0.7602	0.2703	0.3677	Kappa
	Reviews	0.799	0.915	0.715	AUC
		0.735	0.836	0.645	F1
		0.469	0.6712	0.2929	Kappa
	Tweets	0.870	0.794	0.848	AUC
		0.791	0.740	0.737	F1
		0.5824	0.4294	0.4756	Kappa

Table 3: General purpose static embeddings ROC AUC, F1 and Kappa scores from multiple training and testing domains.

		Test domain			
		Word	Reviews	Tweets	
Train domain	Word	0.866	0.785	0.824	AUC
		0.782	0.336	0.359	F1
		0.5672	0.0025	0.0782	Kappa
	Reviews	0.773	0.857	0.828	AUC
		0.701	0.776	0.726	F1
		0.4008	0.5526	0.4553	Kappa
	Tweets	0.741	0.763	0.856	AUC
		0.631	0.590	0.559	F1
		0.2821	0.2451	0.1928	Kappa

Table 4: Edinburgh static embeddings ROC AUC, F1 and Kappa scores for multiple training and testing domains.

[5] https://www.cs.waikato.ac.nz/ml/weka/

In both Table 3 and Table 4, the first column specifies the training domain of the logistic regression. The following columns specify the target domain on which the logistic regression model was tested.

Table 5 shows the results for both the average (AVG) and CLS BERT representations. The first column specifies the training domain, and each consecutive column specifies a testing domain along with the corresponding representation method used in the experiment. For instance, BERT obtained an AUC score of 0.925, a macro-averaged F1 score of 0.848 and a Kappa score of 0.6958 when trained and tested on the word domain using the CLS representation.

		Test domain						
		Word CLS	Word AVG	Reviews CLS	Reviews AVG	Tweets CLS	Tweets AVG	
Train domain	Word CLS	0.925	0.905	0.543	0.612	0.480	0.638	AUC
		0.848	0.818	0.483	0.561	0.260	0.589	F1
		0.6958	0.6368	0.0483	0.1678	-0.0047	0.1957	Kappa
	Word AVG	0.577	0.959	0.500	0.648	0.543	0.807	AUC
		0.543	0.892	0.486	0.572	0.237	0.717	F1
		0.0895	0.7826	-0.0044	0.1995	-0.0008	0.4353	Kappa
	Reviews CLS	0.646	0.913	0.843	0.816	0.617	0.831	AUC
		0.529	0.802	0.770	0.738	0.572	0.725	F1
		0.1586	0.6104	0.5389	0.4787	0.1429	0.4582	Kappa
	Reviews AVG	0.549	0.718	0.483	0.924	0.507	0.759	AUC
		0.275	0.633	0.334	0.845	0.515	0.679	F1
		0.002	0.2718	0.0005	0.69	0.0397	0.3581	Kappa
	Tweets CLS	0.642	0.631	0.509	0.551	0.827	0.561	AUC
		0.566	0.584	0.338	0.486	0.706	0.514	F1
		0.1869	0.1681	0.0032	0.0414	0.4131	0.079	Kappa
	Tweets AVG	0.683	0.899	0.512	0.794	0.555	0.930	AUC
		0.271	0.810	0.458	0.669	0.226	0.836	F1
		0.0003	0.6199	0.0082	0.3573	-0.0003	0.6717	Kappa

Table 5: BERT contextualized embeddings ROC AUC, F1 and Kappa scores from multiple training and testing domains, varying between CLS and AVG representations.

The AUC scores of all transfer learning models also displayed more compactly using bar plots in Figure 6.

Finally, Table 6 shows the winner representation configuration for each task (notice that, except for the results of the diagonal, all results correspond to transfer learning tasks).

5.4 Sentiment Discussion

We will start by discussing results of static word embeddings. We should recall that a good classifier aims to maximize AUC, F1, and Kappa scores.

Both static embeddings exhibit relatively high AUC scores, i.e., Edinburgh embeddings have better performance at the tweet to tweet sentiment classification task

		Test domain			
		Word	Reviews	Tweets	
Train domain	Word	0.959 AVG	0.785 Edin	0.824 Edin	AUC
		0.892 AVG	0.603 GP	0.717 AVG	F1
		0.7826 AVG	0.2703 GP	0.4353 AVG	Kappa
	Reviews	0.913 CLS/AVG	0.924 AVG	0.831 CLS/AVG	AUC
		0.802 CLS/AVG	0.845 AVG	0.726 ED	F1
		0.6104 CLS/AVG	0.69 AVG	0.4582 CLS/AVG	Kappa
	Tweets	0.899 AVG	0.794 GP=AVG	0.930 AVG	AUC
		0.810 AVG	0.740 GP	0.836 AVG	F1
		0.6199 AVG	0.4294 GP	0.6717 AVG	Kappa

Table 6: Winner configuration from all models ROC AUC, F1, and Kappa scores from multiple training and testing domains. AVG represents BERT model with Average to Average embeddings meanwhile CLS/AVG represents BERT with CLS to Average embeddings. GP and Edin refer to Word2Vec General Purpose embeddings and Edinburgh embeddings respectively. An equal symbol between two models indicated that both obtained the same score.

and General purpose embeddings have better performance at movie review to movie review sentiment classification task. From word to movie reviews, both models have a high AUC score. This result indicates that both embeddings have the same ability to extract knowledge from words even though they were trained on different corpora.

Edinburgh embeddings have a better AUC score when transferring from words to tweets. This result is expected as this model was trained over 10 million tweets and has unique tokens for user mentions and URLs, so the test domain has a more reliable representation. Word to tweet transfer results using static embeddings are in line with previous AUC score results obtained by [6]. Meanwhile when transferring from larger domains to the word domain, General purpose embeddings show a better performance.

Almost every high AUC score in General purpose embeddings comes along with high macro-averaged F1 and Kappa scores. Interestingly, for some transfer tasks, Edinburgh embeddings have very low F1 and Kappa scores despite having high AUC scores. A possible explanation for this is that the decision threshold can be shifted when training and testing in different domains.

Our first preliminary experiments using BERT employed the CLS representation for both the training and testing domains. However, some results were inconsistent, as they reported negative Kappa scores and very low F1 scores, such as Word CLS to Tweet CLS. This motivated the evaluation of different ways to represent training and testing domain using BERT. Best results from word domain to word domain are obtained by averaging BERT last hidden layer. This is valid in all other cases when the source and target domains are the same.

A surprising result is that most of the best results were obtained by averaging the last hidden layer of BERT in both training and testing domains, except in movie reviews. Representing movie reviews with the CLS token at the train domain showed very positive results when the testing domain was a different domain represented by

Fig. 6: AUC scores for all models in all sentiment transfer learning tasks. The first 4 bars correspond to all variants of BERT.

the average representation. One possible explanation is that the CLS token somehow manages to synthesize information better when the domain is longer.

Transferring from the word domain to the movie reviews domain is not as good as the other way around. This behavior appears to be recurring when transferring from smaller to longer lexical units. Based on that, we claim that BERT is better extracting contextualized sentiment information from extensive domains and applying this knowledge to smaller domains.

As a general trend, we can claim that BERT's contextualized word embeddings outperform static word embedding models for training and testing over the same domain. This is also true when transferring from a larger to a smaller domain. Word2Vec is a worthy choice when transferring from a smaller to a larger domain.

It is also worth noting that no transfer learning model managed to outperform its counterpart trained on the same domain. This allows us to interpret the diagonal cells of our tables as an upper bound for all transfer learning tasks. We would like to stress that these results should not be perceived as negative. Evidence in transfer learning suggests that, although transfer learning can be a powerful tool for leveraging training data from other related domains, it cannot compete with the standard inductive learning approach based on large training datasets from the target domain [28]. This reveals to us the usefulness of having training examples from the target domain.

However, our study focuses on the scenario where training data for the target domain is absent and cannot be easily acquired. This is a very realistic situation, as many companies or institutions do not have the time or resources to annotate data by affect, but can easily obtain existing training datasets from related domains (e.g., an existing affective lexicon).

Our results indicate that there are many positive transfer learning results that would justify the use of our method in the absence of labeled target domain data. For example, Table 6 shows an AUC score of 0.824 when transferring from words to tweets using static Endinburgh embeddings, an AUC score of 0.913 when transferring from movie reviews to words using the BERT CLS/AVG representation, and an AUC score of 0.831 when transferring from reviews to tweets using BERT CLS/AVG representation. These scores are remarkably high if we consider that AUC scores correspond to the probability of the model scoring a randomly chosen positive example higher than a randomly chosen negative example. These results validate our hypothesis that is possible to leverage affective knowledge between multiple domains by representing textual units with compatible vectors.

5.5 Emotion Experiments

In this subsection, we report results on transferring emotion intensities between words and tweets using static and contextualized word embeddings.

The main difference with previous tasks is that in this case we focus on a regression task. Our dataset covers four different emotions: *anger*, *fear*, *sadness*, and *joy*, and each emotion is continuously rated between 0 and 1 rather than in a discrete space.

General purpose embeddings were not considered in this experiment based on the previous finding that Edinburgh embeddings are more appropriate for Twitter data. We also discard BERT's CLS representation, due to the consistency reported by the average representation in previous experiments. Weka along the LibLINEAR[16] package was used to train a support vector machine regression model, setting the SVMType parameter to L2-regularized L2-loss support vector regression (dual) with the regularization parameter C set to 1.

We consider three metrics to measure the performance of our regression models: Pearson Correlation (COR), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE). Regression results for emotions *anger*, *fear*, *sadness*, and *joy* using both BERT and Edinburgh embeddings are shown in Tables 7, 8, 9, and 10. Analogously to previous experiments, the first column specifies the training domain of the support vector machine regression. Each following column specifies the domain on which the support vector machine regression model was tested along with the representation method used.

Bar plots of the Pearson correlations of all emotion transfer learning experiments are depicted in Figure 7.

		Test Domain				
		Word		Tweet		
		Edin	BERT	Edin	BERT	
Train domain	Word	0.4626	0.6254	0.2526	0.1716	COR
		0.1494	0.136	0.1484	0.2115	MAE
		0.1883	0.1724	0.1863	0.2664	RMSE
	Tweet	0.0938	0.2941	0.4838	0.5816	COR
		0.2003	0.1864	0.1264	0.1215	MAE
		0.2451	0.2296	0.154	0.1516	RMSE

Table 7: Anger transfer learning between word and tweet domains using BERT and Edinburgh (Edin) representations. Metrics shown are Pearson Correlation (COR), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE).

		Test Domain				
		Word		Tweet		
		Edin	BERT	Edin	BERT	
Train domain	Word	0.5053	0.6621	0.2937	0.3032	COR
		0.1429	0.1267	0.1746	0.2138	MAE
		0.1784	0.1631	0.2163	0.2665	RMSE
	Tweet	0.3423	0.4714	0.5435	0.6374	COR
		0.2056	0.1743	0.1439	0.1332	MAE
		0.2574	0.2184	0.1754	0.1674	RMSE

Table 8: Fear transfer learning between word and tweet domains using BERT and Edinburgh (Edin) representations. Metrics shown are Pearson Correlation (COR), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE).

		Test Domain				
		Word		Tweet		
		Edin	BERT	Edin	BERT	
Train domain	Word	0.5156	0.677	0.2812	0.4176	COR
		0.1491	0.125	0.2025	0.2931	MAE
		0.1809	0.157	0.2501	0.3582	RMSE
	Tweet	0.3446	0.4644	0.6013	0.6886	COR
		0.1865	0.2453	0.1454	0.1222	MAE
		0.2313	0.2915	0.1751	0.1535	RMSE

Table 9: Sadness transfer learning between word and tweet domains using BERT and Edinburgh (Edin) representations. Metrics shown are Pearson Correlation (COR), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE).

		Test Domain				
		Word		Tweet		
		Edin	BERT	Edin	BERT	
Train domain	Word	0.5966	0.5653	0.3587	0.3373	COR
		0.1359	0.1478	0.1789	0.3322	MAE
		0.1675	0.1873	0.2148	0.3912	RMSE
	Tweet	0.4001	0.403	0.5708	0.6146	COR
		0.2256	0.1995	0.1534	0.1421	MAE
		0.2705	0.2502	0.1864	0.1828	RMSE

Table 10: Joy transfer learning between word and tweet domains using BERT and Edinburgh (Edin) representations. Metrics shown are Pearson Correlation (COR), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE).

Fig. 7: Pearson correlation for all models in all emotion intensity transfer learning tasks.

5.6 Emotion Discussion

It is important to note that in this task the goal is to minimize the values of MAE and RMSE, while maximizing the value of COR. Interestingly, predictive performance differs significantly between different emotions for Edinburgh embeddings. When source and target domains are the same, correlations for *joy* are as high as 0.5966 and 0.5708 against values of 0.4626 and 0.4838 obtained for *anger*. This may be caused by Edinburgh embeddings not being able to distinguish *anger* with the same accuracy as *joy* from the embedding space. Another unexpected result occurs when transferring from tweet to words using Edinburgh embeddings. Somehow, these embeddings fail at transferring *anger*, obtaining a correlation of 0.0938, opposed to 0.4001 obtained for *joy*. This result suggests that there is a substantial discordance in the way *anger* is expressed in tweets and words. This will be further analyzed in Subsection 5.7.

In relation to emotion experiments using BERT, the worst results are obtained when transferring from domains with shorter lexical units than those of the target domain, similarly to what is reported in the sentiment experiments. This is expected, as BERT's self-attention mechanism excels at extracting contextual information from larger contexts. This model performs well in general, except when transferring *anger* from tweets to words. This emotion seems to be difficult to capture for both representation models.

If we compare both representation approaches, we can conclude that each approach obtains similar results to those from the sentiment experiments, with BERT extracting better knowledge from domains with longer lexical units to shorter ones. BERT also dominates the diagonal, so it achieves better predictions within the same domains. Meanwhile, Edinburgh embeddings exhibit a slightly better performance than BERT when transferring from words to tweets. This behavior seems to be repeated in the other emotions: BERT dominating all transfer learning experiments excepting the word to tweet task, in which the Edinburgh embeddings are competitive.

These correlations are in line with previous results obtained in [32]. We must remark that we are using BERT's base model only as a feature extractor mechanism. A fine-tuned version on the test domain would very likely perform better in a same-domain train-test evaluation setting. However, in our transfer learning setting we are not allowing the model to access labeled examples from the target domain during training. It is unclear whether fine-tuning on the train domain would lead to improvements when the test domain is different, which is the main objective of this work.

5.7 Qualitative Analysis

In this subsection we perform a qualitative analysis of the classification errors made by our transfer learning models with the aim understanding the limitations of our approach.

Table 11 shows some examples of tweets that were misclassified by all models in the task of transferring sentiment from words to tweets.

Actual	Predicted	Tweet
positive	negative	Gas by my house hit $3.39!!!! I'm going to Chapel Hill on Sat. :)
positive	negative	with J Davlar 11th. Main rivals are team Poland. Hopefully we can make it a successful end to a tough week of training tomorrow.
positive	negative	Never start working on your dreams and goals tomorrow..... tomorrow never comes....if it means anything to U, ACT NOW! #getafterit

Table 11: Examples of classification errors of both models in tweet prediction, trained on words.

All these tweets contain words with strong negative sentiment, such as "tough", "rivals", and "never". We can conclude that the presence of specific negative words can mislead models that were trained on the word domain.

As for the task of detecting the intensity of emotions, the results showed that anger was particularly difficult to transfer from the tweet domain to the word domain. To get further insight on this, we extracted the 10 words with the largest absolute prediction error for both BERT and Edinburgh based models, which are shown in Tables 12 and 13 respectively.

Actual	Predicted	Error	Word
0.818	0.293	-0.525	terrorist
0.562	0.037	-0.525	torpedo
0.939	0.41	-0.529	terrorize
0.844	0.308	-0.536	tumultuous
0.621	0.084	-0.537	theft
0.859	0.313	-0.546	tirade
0.851	0.295	-0.556	terrorism
0.825	0.26	-0.565	ruinous
0.544	-0.028	-0.572	robbery
0.862	0.227	-0.635	smite

Table 12: Prediction errors of BERT AVG representation at word anger prediction, trained on tweets.

It is interesting to observe that the BERT model (trained on tweets) fails at predicting the anger intensity of three words derived from the same root "terror": "terrorist", "terrorize", and "terrorism". When examining the training tweets, we noticed that none of them contains words with that root. This suggests that the model did not receive enough evidence to adequately learn the affect of these word, which resulted in predicting lower intensities than expected.

In the case of Edinburgh, we did not notice such a clear pattern as with BERT.

We randomly took the word "unhelpful" from that list and analyzed the training tweets that contain it. Some examples of tweets containing mentions of that word are shown in Table 14.

Actual	Predicted	Error	Word
0.125	0.653	0.528	wireless
0.844	0.315	-0.529	tumultuous
0.781	0.251	-0.53	sinister
0.885	0.353	-0.532	wrath
0.03	0.57	0.54	waffle
0.219	0.765	0.546	unhelpful
0.152	0.715	0.563	underpaid
0.814	0.235	-0.579	savage
0.219	0.821	0.602	whiny
0.328	0.95	0.622	spammers

Table 13: Prediction errors of Word2Vec Edinburgh representation at word anger prediction, trained on tweets.

Training tweets	Anger
@ThomsonCares Sam- yes we have! Not helpful at all! We need this sorting ASAP! You keep promising stuff that doesnt happen!!!! #fuming	0.771
@lynnew69 then he said talking about wills uncontrollable animals when moving to another link. These comments do not help! #fuming	0.75
Zero help from @ups customer service. Just pushing the buck back and forth and promising callbacks that don't happen. #anger #loathing	0.854

Table 14: Tweets with the presence of words related to "unhelpful"

We observe that the predicted value of word "unhelpful" is 0.765, while the value delivered by the human annotators is 0.219. Tweets containing the word "unhelpful" have an intensity greater than 0.7 which is closer to the model's prediction. This indicates that for certain words there is no direct correspondence between the contextual anger intensity expressed in a tweet and the isolated intensity of the word.

As a result of this qualitative analysis we can conclude the following:

1. When transferring from words to tweets, the presence of words with the opposite sentiment than the target tweet can mislead the prediction.
2. When transferring from tweets to words, the absence of training tweets containing the target word may prevent its accurate prediction.
3. The affect intensity conveyed by an isolated word can be different than its contextualized intensity in a sentence.

6 Conclusions

This paper has presented a novel method for leveraging affect knowledge between three different domains: movie reviews, tweets, and words.

Our method exploits the fact that despite the apparent differences between these domains, the sentiment and emotion label spaces are shared across them (i.e., texts of different length can plausibly be mapped to the same affect labels). This can be

particularly useful when training data is not available for the target domain for which the affect analysis is intended.

We also exploit the property that both static and contextual word embeddings can be aggregated to represent textual units of different lengths (e.g., movie reviews, tweets, words) as compatible vectors. Consequently, a classifier trained with data from one source domain can easily be applied to data from a different target domain.

Our results indicate that, in general, affect knowledge can be transferred between one domain to another using our method. However, classification performance can vary significantly depending on the choice of source-target domain pair and the representation method. Word2Vec tends to produce more stable results than BERT and performs relatively well in many transfer learning tasks. Word2Vec vectors trained on Twitter data work significantly better than General purpose embeddings for tweets' sentiment classification. Concerning BERT, we observe that BERT-derived representations can outperform WordVec in many tasks. However, these results exhibit more variability depending on the aggregation approach used.

Another remarkable result is that, in many cases, the transfer classification results show high scores for the area under the ROC curve (AUC) metric and low scores for F1 and Kappa. This anomaly suggests that the decision boundary gets shifted when moving from one sentiment domain to another. This problem could be mitigated by adjusting the decision threshold on the target domain.

The emotion experiments results suggest that *anger* intensity detection is more challenging than *joy*, *sadness*, and *fear* when transferring between word and tweet domains. This decrease is caused by a mismatch in how anger is perceived in tweets and single words.

Finally, as a general trend, we observe that affect knowledge can be easier transferred from longer to shorter domains (e.g., movie reviews to tweets or tweets to words) than the opposite way. We attribute this to the fact that the training domain is richer in contextual affective information in those cases. The main contribution of this paper is a new method to leverage affective labels between diverse domains. This approach can be especially useful for practitioners who lack the resources for creating annotated data for their target domain.

We envision several avenues of future work. First, we plan to explore our method with other affect labels, such as the Hourglass of Emotions [50] and hate speech. Second, we will study how to incorporate other recently developed contextualized models such as XLNet [57], RoBERTa [24], and ERNIE [59] into our method.

Compliance with Ethical Standards

Funding: This work was funded by ANID FONDECYT grant 11200290, U-Inicia VID Project UI-004/20 and ANID - Millennium Science Initiative Program - Code ICN17_002.
Conflict of Interest: The authors declare that they have no conflict of interest.
Informed Consent: Informed consent was not required as no human or animals were involved.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Akhtar, M.S., Ekbal, A., Cambria, E.: How intense are you? predicting intensities of emotions and sentiments using stacked ensemble [application notes]. IEEE Computational Intelligence Magazine **15**(1), 64–75 (2020)
2. Amir, S., Astudillo, R., Ling, W., Martins, B., Silva, M.J., Trancoso, I.: Inesc-id: A regression model for large scale twitter sentiment lexicon induction. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pp. 613–618 (2015)
3. Behdenna, S., Barigou, F., Belalem, G.: Sentiment analysis at document level. In: A. Unal, M. Nayak, D.K. Mishra, D. Singh, A. Joshi (eds.) Smart Trends in Information Technology and Computer Communications, pp. 159–168. Springer Singapore, Singapore (2016)
4. Bravo-Marquez, F., Frank, E., Mohammad, S.M., Pfahringer, B.: Determining word-emotion associations from tweets by multi-label classification. In: 2016 IEEE/WIC/ACM International Conference on Web Intelligence, WI 2016, Omaha, NE, USA, October 13-16, 2016, pp. 536–539. IEEE Computer Society (2016). DOI 10.1109/WI.2016.0091. URL `https://doi.org/10.1109/WI.2016.0091`
5. Bravo-Marquez, F., Frank, E., Pfahringer, B.: From unlabelled tweets to twitter-specific opinion words. In: R. Baeza-Yates, M. Lalmas, A. Moffat, B.A. Ribeiro-Neto (eds.) Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015, pp. 743–746. ACM (2015). DOI 10.1145/2766462.2767770. URL `https://doi.org/10.1145/2766462.2767770`
6. Bravo-Marquez, F., Frank, E., Pfahringer, B.: Transferring sentiment knowledge between words and tweets. Web Intelligence **16**(4), 203–220 (2018). DOI 10.3233/WEB-180389. URL `https://doi.org/10.3233/WEB-180389`
7. Camacho-Collados, J., Pilehvar, M.T.: From word to sense embeddings: A survey on vector representations of meaning. J. Artif. Intell. Res. **63**, 743–788 (2018). DOI 10.1613/jair.1.11259. URL `https://doi.org/10.1613/jair.1.11259`
8. Cambria, E.: Affective computing and sentiment analysis. IEEE intelligent systems **31**(2), 102–107 (2016)
9. Cambria, E., Hussain, A.: Sentic computing. Cognitive Computation **7**(2), 183–185 (2015)
10. Cambria, E., Li, Y., Xing, F.Z., Poria, S., Kwok, K.: Senticnet 6: Ensemble application of symbolic and subsymbolic ai for sentiment analysis. CIKM'20, Oct 20-24 (2020)
11. Cañete, J., Chaperon, G., Fuentes, R., Ho, J.H., Kang, H., Pérez, J.: Spanish pre-trained bert model and evaluation data. In: Practical ML for Developing Countries Workshop@ ICLR 2020 (2020)
12. Chalil, R.P., Selvaraju, S., Mahalakshmi, G.S.: Twitter sentiment analysis for large-scale data: An unsupervised approach. Cogn. Comput. **7**(2), 254–262 (2015). DOI 10.1007/s12559-014-9310-z. URL `https://doi.org/10.1007/s12559-014-9310-z`
13. Clark, K., Luong, M., Le, Q.V., Manning, C.D.: ELECTRA: pre-training text encoders as discriminators rather than generators. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net (2020). URL `https://openreview.net/forum?id=r1xMH1BtvB`
14. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: J. Burstein, C. Doran, T. Solorio (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pp. 4171–4186. Association for Computational Linguistics (2019). DOI 10.18653/v1/n19-1423. URL `https://doi.org/10.18653/v1/n19-1423`
15. Esuli, A., Sebastiani, F.: Determining the semantic orientation of terms through gloss classification. In: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, pp. 617–624. ACM (2005)
16. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: Liblinear: A library for large linear classification. J. Mach. Learn. Res. **9**, 1871–1874 (2008)

17. Glorot, X., Bordes, A., Bengio, Y.: Domain adaptation for large-scale sentiment classification: A deep learning approach. In: L. Getoor, T. Scheffer (eds.) Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011, pp. 513–520. Omnipress (2011). URL https://icml.cc/2011/papers/342_icmlpaper.pdf
18. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. ACM SIGKDD explorations newsletter **11**(1), 10–18 (2009)
19. Harris, Z.S.: Distributional structure. Word **10**(2-3), 146–162 (1954)
20. He, R., Lee, W.S., Ng, H.T., Dahlmeier, D.: Exploiting document knowledge for aspect-level sentiment classification. CoRR **abs/1806.04346** (2018). URL http://arxiv.org/abs/1806.04346
21. Joyce, B., Deng, J.: Sentiment analysis of tweets for the 2016 us presidential election. In: 2017 IEEE MIT Undergraduate Research Technology Conference (URTC), pp. 1–4 (2017)
22. Kim, Y.: Convolutional neural networks for sentence classification. In: A. Moschitti, B. Pang, W. Daelemans (eds.) Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pp. 1746–1751. ACL (2014). DOI 10.3115/v1/d14-1181. URL https://doi.org/10.3115/v1/d14-1181
23. Liu, B.: Sentiment analysis and opinion mining. Synthesis lectures on human language technologies **5**(1), 1–167 (2012)
24. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized BERT pretraining approach. CoRR **abs/1907.11692** (2019). URL http://arxiv.org/abs/1907.11692
25. Ma, Y., Peng, H., Khan, T., Cambria, E., Hussain, A.: Sentic LSTM: a hybrid network for targeted aspect-based sentiment analysis. Cogn. Comput. **10**(4), 639–650 (2018). DOI 10.1007/s12559-018-9549-x. URL https://doi.org/10.1007/s12559-018-9549-x
26. Maas, A.L., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: D. Lin, Y. Matsumoto, R. Mihalcea (eds.) The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA, pp. 142–150. The Association for Computer Linguistics (2011). URL https://www.aclweb.org/anthology/P11-1015/
27. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
28. McCann, B., Keskar, N.S., Xiong, C., Socher, R.: The natural language decathlon: Multitask learning as question answering. arXiv preprint arXiv:1806.08730 (2018)
29. Melville, P., Gryc, W., Lawrence, R.D.: Sentiment analysis of blogs by combining lexical knowledge with text classification. In: J.F.E. IV, F. Fogelman-Soulié, P.A. Flach, M.J. Zaki (eds.) Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009, pp. 1275–1284. ACM (2009). DOI 10.1145/1557019.1557156. URL https://doi.org/10.1145/1557019.1557156
30. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: C.J.C. Burges, L. Bottou, Z. Ghahramani, K.Q. Weinberger (eds.) Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States, pp. 3111–3119 (2013). URL http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality
31. Miller, G.A., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.: Wordnet: An on-line lexical database. International Journal of Lexicography **3**, 235–244 (1990)
32. Mohammad, S., Bravo-Marquez, F.: WASSA-2017 shared task on emotion intensity. In: Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, pp. 34–49. Association for Computational Linguistics, Copenhagen, Denmark (2017). DOI 10.18653/v1/W17-5205. URL https://www.aclweb.org/anthology/W17-5205
33. Mohammad, S., Turney, P.D.: Crowdsourcing a word-emotion association lexicon. Comput. Intell. **29**(3), 436–465 (2013). DOI 10.1111/j.1467-8640.2012.00460.x. URL https://doi.org/10.1111/j.1467-8640.2012.00460.x
34. Mohammad, S.M.: Word affect intensities. In: Proceedings of the 11th Edition of the Language Resources and Evaluation Conference (LREC-2018). Miyazaki, Japan (2018)
35. Nguyen, H., Nguyen, M.: A deep neural architecture for sentence-level sentiment classification in twitter social networking. In: K. Hasida, W.P. Pa (eds.) Computational Linguistics - 15th International Conference of the Pacific Association for Computational Linguistics, PACLING 2017, Yangon,

Myanmar, August 16-18, 2017, Revised Selected Papers, *Communications in Computer and Information Science*, vol. 781, pp. 15–27. Springer (2017). DOI 10.1007/978-981-10-8438-6_2. URL https://doi.org/10.1007/978-981-10-8438-6_2

36. Nielsen, F.Å.: A new ANEW: evaluation of a word list for sentiment analysis in microblogs. In: M. Rowe, M. Stankovic, A. Dadzie, M. Hardey (eds.) Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages, Heraklion, Crete, Greece, May 30, 2011, *CEUR Workshop Proceedings*, vol. 718, pp. 93–98. CEUR-WS.org (2011). URL http://ceur-ws.org/Vol-718/paper_16.pdf
37. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2010). DOI 10.1109/TKDE.2009.191. URL https://doi.org/10.1109/TKDE.2009.191
38. Pelletier, F.J.: The principle of semantic compositionality. Topoi **13**(1), 11–24 (1994)
39. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: A. Moschitti, B. Pang, W. Daelemans (eds.) Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pp. 1532–1543. ACL (2014). DOI 10.3115/v1/d14-1162. URL https://doi.org/10.3115/v1/d14-1162
40. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: M.A. Walker, H. Ji, A. Stent (eds.) Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers), pp. 2227–2237. Association for Computational Linguistics (2018). DOI 10.18653/v1/n18-1202. URL https://doi.org/10.18653/v1/n18-1202
41. Petrović, S., Osborne, M., Lavrenko, V.: The Edinburgh twitter corpus. In: Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media, pp. 25–26. Association for Computational Linguistics, Los Angeles, California, USA (2010). URL https://www.aclweb.org/anthology/W10-0513
42. Poria, S., Cambria, E., Winterstein, G., Huang, G.: Sentic patterns: Dependency-based rules for concept-level sentiment analysis. Knowl. Based Syst. **69**, 45–63 (2014). DOI 10.1016/j.knosys.2014.05.005. URL https://doi.org/10.1016/j.knosys.2014.05.005
43. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training. URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf (2018)
44. Rani, S., Kumar, P.: Deep learning based sentiment analysis using convolution neural network. Arabian Journal for Science and Engineering **44**(4), 3305–3314 (2019)
45. Rosenthal, S., Ritter, A., Nakov, P., Stoyanov, V.: Semeval-2014 task 9: Sentiment analysis in twitter. In: P. Nakov, T. Zesch (eds.) Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014, pp. 73–80. The Association for Computer Linguistics (2014). DOI 10.3115/v1/s14-2009. URL https://doi.org/10.3115/v1/s14-2009
46. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. CoRR **abs/1910.01108** (2019). URL http://arxiv.org/abs/1910.01108
47. Sindhwani, V., Melville, P.: Document-word co-regularization for semi-supervised sentiment analysis. In: Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy, pp. 1025–1030. IEEE Computer Society (2008). DOI 10.1109/ICDM.2008.113. URL https://doi.org/10.1109/ICDM.2008.113
48. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL, pp. 1631–1642. ACL (2013). URL https://www.aclweb.org/anthology/D13-1170/
49. Suhariyanto, Firmanto, A., Sarno, R.: Prediction of movie sentiment based on reviews and score on rotten tomatoes using sentiwordnet. In: 2018 International Seminar on Application for Technology of Information and Communication, pp. 202–206 (2018)
50. Susanto, Y., Livingstone, A.G., Ng, B.C., Cambria, E.: The hourglass model revisited. IEEE Intelligent Systems **35**(5), 96–102 (2020). DOI 10.1109/MIS.2020.2992799
51. Trinh, T.H., Dai, A.M., Luong, T., Le, Q.V.: Learning longer-term dependencies in rnns with auxiliary losses. CoRR **abs/1803.00144** (2018). URL http://arxiv.org/abs/1803.00144

52. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: I. Guyon, U. von Luxburg, S. Bengio, H.M. Wallach, R. Fergus, S.V.N. Vishwanathan, R. Garnett (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA, pp. 5998–6008 (2017). URL http://papers.nips.cc/paper/7181-attention-is-all-you-need
53. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity in phrase-level sentiment analysis. In: HLT/EMNLP 2005, Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, 6-8 October 2005, Vancouver, British Columbia, Canada, pp. 347–354. The Association for Computational Linguistics (2005). URL https://www.aclweb.org/anthology/H05-1044/
54. Wu, N., Green, B., Ben, X., O'Banion, S.: Deep transformer models for time series forecasting: The influenza prevalence case. CoRR **abs/2001.08317** (2020). URL https://arxiv.org/abs/2001.08317
55. Xia, Y., Cambria, E., Hussain, A., Zhao, H.: Word polarity disambiguation using bayesian model and opinion-level features. Cognitive Computation **7** (2014). DOI 10.1007/s12559-014-9298-4
56. Yang, Z., Dai, Z., Yang, Y., Carbonell, J.G., Salakhutdinov, R., Le, Q.V.: Xlnet: Generalized autoregressive pretraining for language understanding. In: H.M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E.B. Fox, R. Garnett (eds.) Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada, pp. 5754–5764 (2019). URL http://papers.nips.cc/paper/8812-xlnet-generalized-autoregressive-pretraining-for-language-understanding
57. Yang, Z., Dai, Z., Yang, Y., Carbonell, J.G., Salakhutdinov, R., Le, Q.V.: Xlnet: Generalized autoregressive pretraining for language understanding. In: H.M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E.B. Fox, R. Garnett (eds.) Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada, pp. 5754–5764 (2019). URL http://papers.nips.cc/paper/8812-xlnet-generalized-autoregressive-pretraining-for-language-understanding
58. Yu, L., Wang, J., Lai, K.R., Zhang, X.: Refining word embeddings for sentiment analysis. In: M. Palmer, R. Hwa, S. Riedel (eds.) Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017, pp. 534–539. Association for Computational Linguistics (2017). DOI 10.18653/v1/d17-1056. URL https://doi.org/10.18653/v1/d17-1056
59. Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., Liu, Q.: ERNIE: Enhanced language representation with informative entities. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 1441–1451. Association for Computational Linguistics, Florence, Italy (2019). DOI 10.18653/v1/P19-1139. URL https://www.aclweb.org/anthology/P19-1139
60. Zhu, Y., Kiros, R., Zemel, R.S., Salakhutdinov, R., Urtasun, R., Torralba, A., Fidler, S.: Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. CoRR **abs/1506.06724** (2015). URL http://arxiv.org/abs/1506.06724
61. Zou, W.Y., Socher, R., Cer, D.M., Manning, C.D.: Bilingual word embeddings for phrase-based machine translation. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL, pp. 1393–1398. ACL (2013). URL https://www.aclweb.org/anthology/D13-1141/