Decoding Sentiment from Distributed Representations of Sentences

Edoardo Maria Ponti
ep490@cam.ac.uk

Ivan Vulić
iv250@cam.ac.uk

Anna Korhonen
alk23@cam.ac.uk

Language Technology Lab, University of Cambridge

Abstract

Distributed representations of sentences have been developed recently to represent their meaning as real-valued vectors. However, it is not clear how much information such representations retain about the polarity of sentences. To study this question, we decode sentiment from unsupervised sentence representations learned with different architectures (sensitive to the order of words, the order of sentences, or none) in 9 typologically diverse languages. Sentiment results from the (recursive) composition of lexical items and grammatical strategies such as negation and concession. The results are manifold: we show that there is no 'one-size-fits-all' representation architecture outperforming the others across the board. Rather, the top-ranking architectures depend on the language and data at hand. Moreover, we find that in several cases the additive composition model based on skip-gram word vectors may surpass supervised state-of-art architectures such as bidirectional LSTMs. Finally, we provide a possible explanation of the observed variation based on the type of negative constructions in each language.

1 Introduction

Distributed representations of sentences are usually acquired in an unsupervised fashion from raw texts. Those inferred from different algorithms are prone to grasp parts of their meaning and disregard others. Representations have been evaluated thoroughly, both intrinsically (interpretation through distance measures) and extrinsically (performance on downstream tasks). Moreover, several methods have been considered, based on both the composition of word embeddings (Milajevs et al., 2014; Marelli et al., 2014; Sultan et al., 2015) and direct generation (Hill et al., 2016). The evaluation was focused solely on English, and it rarely concerned other languages (Adi et al., 2017; Conneau et al., 2017). As a consequence, many 'core' methods to learn distributed sentence representations are largely under-explored in a variety of typologically diverse languages, and still lack a demonstration of their usefulness in actual downstream tasks.

In this work, we study how well distributed sentence representations capture the *polarity of a sentence*. To this end, we choose the Sentiment Analysis task as an extrinsic evaluation protocol: it directly detects the polarity of a text, where polarity is defined as the attitude of the speaker with respect to the whole content of the string or one of the entities mentioned therein. This attitude is measured quantitatively on a scale spanning from negative to positive with arbitrary granularity. As such, polarity consists in a crucial part of the meaning of a sentence, which should not be lost.

The polarity of a sentence depends heavily on a complex interaction between lexical items endowed with an intrinsic polarity, and morphosyntactic constructions altering polarity, most notably negation and concession. The interaction is deemed to be recursive, hence some approaches take into account word order and phrase boundaries in order to apply the correct composition (Socher et al., 2013). However, some languages lack continuous constituents: contiguous spans of words do not correspond to syntactic subtrees, making composition unreliable (Ponti, 2016). Moreover, the expression of negation varies across languages, as demonstrated by works in Linguistic Typology (Dahl, 1979, *inter alia*). In particular, negation can appear as a bounded morpheme or a free morpheme; it can precede or follow the verb; it can 'agree' or not in polarity with indefinite pronouns; it can alter the expression of verbal categories (e.g. tense, aspect, or modality).

We explore a series of methods endowed with different features: some hinge upon word order, others on sentence order, others on neither. We evaluate these unsupervised representations using a Multi-Layer Perceptron which uses the generated sentence representations as input and predicts sentiment classes (positive vs. negative) as output. Training and evaluation are based on a collection of annotated databases. Owing to the variety of methods and languages, we expect to observe a variation in the performance correlated with the properties of both.

Moreover, we establish a ceiling to the possible performances of our method based on decoding unsupervised distributed representations. In fact, we offer a comparison between this and supervised deep learning architectures that achieve state-of-art scores in the Sentiment Analysis task. In particular, we also evaluate a bi-directional LSTM (Li et al., 2015) on the same task. These models have advantage over distributed representations as: i) they are specialised on a single task rather than built as general-purpose representations; ii) their recurrent nature allows to capture the sequential composition of polarity in a sentence. However, since training these models requires large amounts of annotated data, resource scarcity in other languages hampers their portability.

The aim of this work is to assess which algorithm for distributed sentence representations is the most appropriate for capturing polarity in a given language. Moreover, we study how language-specific properties have an impact on performance, finding an explanation in Language Typology. We also provide an in-depth analysis of the most relevant features by visualising the activation of hidden neurons. This will hopefully contribute to advancing the Sentiment Analysis task in the multilingual scenarios. In § 2, we survey prior work on multilingual sentiment analysis. Afterwards, we present the tested algorithms for generating distributed representations of sentences in § 3. In § 4, we sketch the dataset and the experimental setup. Finally, § 5 examines the results in light of the sensitivity of the algorithms and the typology of negation.

2 Multilingual Sentiment Analysis

The task of sentiment classification is mostly addressed through supervised approaches. However, these achieve unsatisfactory results in resource-lean languages because of the scarcity of resources to train dedicated models (Denecke, 2008). This afflicts state-of-art deep learning architectures even more compared to traditional machine learning algorithms (Chen et al., 2016). As a consequence, previous work resorted to i) language transfer or ii) joint multilingual learning. The former adapts models from a source resource-rich language to a target resource-poor language; the latter infers a single model portable across languages. Approaches based on distributed representations induced in an unsupervised fashion do not face the difficulty resulting from resource scarcity: they are portable to other tasks and languages. In this section we survey deep learning techniques, adaptive models, and unsupervised distributed representations for sentiment classification in a multilingual scenario. The last approach is the focus of this work.

Deep learning algorithms for sentiment classification are designed to deal with compositionality. Hence, they often rely on recurrent networks tracing the sequential history of a sentence, or special compositional devices. Recurrent models include bi-directional LSTMs (Li et al., 2015), possibly enriched with context (Mousa and Schuller, 2017). On the other hand, Socher et al. (2013) put forth a Recursive Neural Tensor Network, which composes representations recursively through a single tensor-based composition function. Subsequent improvements of this line of research include the Structural Attention Neural Networks (Kokkinos and Potamianos, 2017), which adds structural information around each node of a syntactic tree.

When supervised monolingual models are not feasible, language transfer can bridge between multiple languages, for instance through supervised latent Dirichlet allocation (Boyd-Graber and Resnik, 2010). Direct transfer relies on word-aligned parallel texts where the source language text is either manually or automatically annotated. The sentiment information is then projected onto the target text (Almeida et al., 2015), also leveraging non-parallel data (Zhou et al., 2015). Chen et al. (2016) devised a multi-task network where an adversarial branch spurs the shared layers to learn language-independent features. Finally, Lu et al. (2011) learned from annotated examples in both the source and target language. Alternatively, sentences from other languages are translated into English and assigned a sentiment based on lexical resources (Denecke, 2008) or supervised methods

(Balahur and Turchi, 2014).

Finally, cross-lingual sentiment classification can leverage on shared distributed representations. Zhou et al. (2016) captured shared high-level features across aligned sentences through autoencoders. In this latent space, distances were optimised to reflect differences in sentiment. On the other hand, Fernández et al. (2015) exploited bilingual word representations, where vector dimensions mirror the distributional overlap with respect to a pivot. Le and Mikolov (2014) concatenated sentence representations obtained through variants of Paragraph Vector and trained a Logistic Regression model on top of them.

Previous studies thus demonstrated that sentence representations retain information about polarity, and that they partly alleviate the drawbacks of deep architectures (single-purposed and data-demanding). Hence, the Sentiment Analysis tasks seems convenient to compare different sentence representation architectures. Nonetheless, a systematic evaluation has never taken place for this task, and a large-scale study over typologically diverse languages has not been attempted for any of the algorithms reviewed. We intend to fill these gaps, considering the methods to generate sentence representations outlined in the next section.

3 Distributed Sentence Representations

Word vectors can be combined through various compositional operations to obtain representations of phrases and sentences. Mitchell and Lapata (2010) explored two operations: addition and multiplication. Notwithstanding their simplicity, they are hardly outperformed by more sophisticated operations (Rimell et al., 2016). Some of these compositional representations based on matrix multiplication were also evaluated on sentiment classification (Yessenalina and Cardie, 2011). Alternatively, sentence representations can be induced directly with no intermediate step at the word level. In this paper, we focus on sentence representations that are generated in an unsupervised fashion. Furthermore, they are 'fixed', that is, they are not fine-tuned for any particular downstream task, since we are interested in their intrinsic content.[1]

3.1 Algorithms

We explore several methods to generate sentence representations. One exploits a compositional operation (addition) over word representations stemming from a Skip-Gram model (§ 3.1.1). Others are direct methods, including FastSent (§ 3.1.2), a Sequential Denoising AutoEncoder (SDAE, § 3.1.3) and Paragraph Vector (§ 3.1.4). Note that FastSent relies on sentence order, SDAE on word order, and Paragraph Vector on neither. All these algorithms were trained on cleaned-up Wikipedia dumps.

The choice of the algorithms was based on following criteria: i) their performance reported in recent surveys (n.b., the surveys were limited to English and evaluated on other tasks), most notably Hill et al. (2016) and Milajevs et al. (2014); ii) the variety of their modelling assumptions and features encoded. The referenced surveys already hinted that the usefulness of a representation is largely dependent on the actual application. Shallower but more interpretable representations can be decoded with spatial distance metrics. Others, more deep and convoluted architectures, outperform the others in supervised tasks. We inquire whether the generalisation is tenable also in the task of Sentiment Analysis targeting sentence polarity.

3.1.1 Additive Skip-Gram

As a bottom-up method, we train word embeddings using skip-gram with negative sampling (Mikolov et al., 2013). The algorithm finds the parameter θ such that, given a pair of a word w and a context c, the model discriminates correctly whether it belongs to a set of sentences S or a set of randomly generated incorrect sentences S':

$$\prod_{(w,c)\in S} p(S=1|w,c,\theta) \prod_{(w,c)\in S'} p(S'=0|w,c,\theta)$$

The representation of a sentence was obtained via element-wise addition of the vectors of the words belonging to it (Mitchell and Lapata, 2010).

3.1.2 FastSent

The FastSent model was proposed by Hill et al. (2016). It hinges on a sentence-level distributional hypothesis (Polajnar et al., 2015; Kiros et al., 2015). In other terms, it assumes that the meaning of a sentence can be inferred by the neighbour sentences in a text. It is a simple additive log-linear model conceived to mitigate the computational expensiveness of algorithms based on a similar assumption.

[1] This excludes methods concerned with phrases, like the ECO embeddings (Poliak et al., 2017), or requiring structured knowledge, like CHARAGRAM (Wieting et al., 2016a).

Hence, it was preferred over SkipThought (Kiros et al., 2015) because of i) these efficiency issues and ii) its competitive performances reported by Hill et al. (2016). In FastSent, sentences are represented as bags of words: a context of sentences is used to predict the adjacent sentence. Each word *w* corresponds to a source vector u_w and a target vector v_w. A sentence S_i is represented as the sum of the source vectors of its words $\sum_{w \in S_i} u_w$. Hence, the cost *C* of a representation is given by the softmax $\sigma(x)$ of a sentence representation and the target vectors of the words in its context *c*.

$$C_{S_i} = \sum_{c \in S_{i-1} \cup S_{i+1}} \sigma(\sum_{w \in S_i} u_w, v_c) \qquad (1)$$

This model does not rely on word order, but rather on sentence order. It encodes new sentences by summing over the source vectors of their words.

3.1.3 Sequential Denoising AutoEncoder

Sequential Denoising AutoEncoders (SDAEs) combine features of Denoising AutoEncoders (DAE) and Sequence-to-Sequence models. In DAE, the input representation is corrupted by a noise function and the algorithms learns to recover the original (Vincent et al., 2008). Intuitively, this makes the model more robust to changes in input that are irrelevant for the task at hand. This architecture was later adapted to encode and decode variable-length inputs, and the corruption process was implemented in the form of dropout (Iyyer et al., 2015). In the implementation by Hill et al. (2016),[2] the corruption function is defined as $f(S|p_o, p_x)$. S is a list of words (a sentence) where each has a probability p_o to be deleted, and the order of the words in every distinct bigram has a probability p_x to be swapped. The architecture consists in a Recurrent Layer and predicts $p(S|f(S|p_o, p_x))$.

3.1.4 Paragraph Vector

Paragraph Vector is a collection of log-linear models proposed by Le and Mikolov (2014) for paragraph/sentence representation. It consists of two different models, namely the Distributed Memory model (DM) and the Distributed Bag Of Words model (DBOW). In DM, the ID of every distinct paragraph (or sentence) is mapped to a unique vector in a matrix *D* and each word is mapped to a unique vector in matrix *W*. Given a sentence *i* and a window size *k*, the vector $D_{i,\cdot}$ is used in conjunction with the concatenation of the vectors of the words in a sampled context $\langle w_{i_1}, \dots, w_{i_k} \rangle$ to predict the next word through logistic regression:

$$p(W_{i_{k+1}} | \langle D_i, W_{i_1}, \dots, W_{i_k} \rangle) \qquad (2)$$

Note that the sentence ID vector is shared by the contexts sampled from the same sentence. On the other hand, DBOW focuses on predicting the word embedding W_{i_j} for a sampled word *j* belonging to sentence *i* given the sentence representation D_i. As a result, the main difference between the two Paragraph Vector models is that the first is sensitive to word order (represented by the word vector concatenation), whereas the second is insensitive with respect to it. These models store a representation for each sentence in the training set, hence they are memory demanding. We use the *gensim* implementation of the two models available as Doc2Vec.[3]

3.2 Hyper-parameters

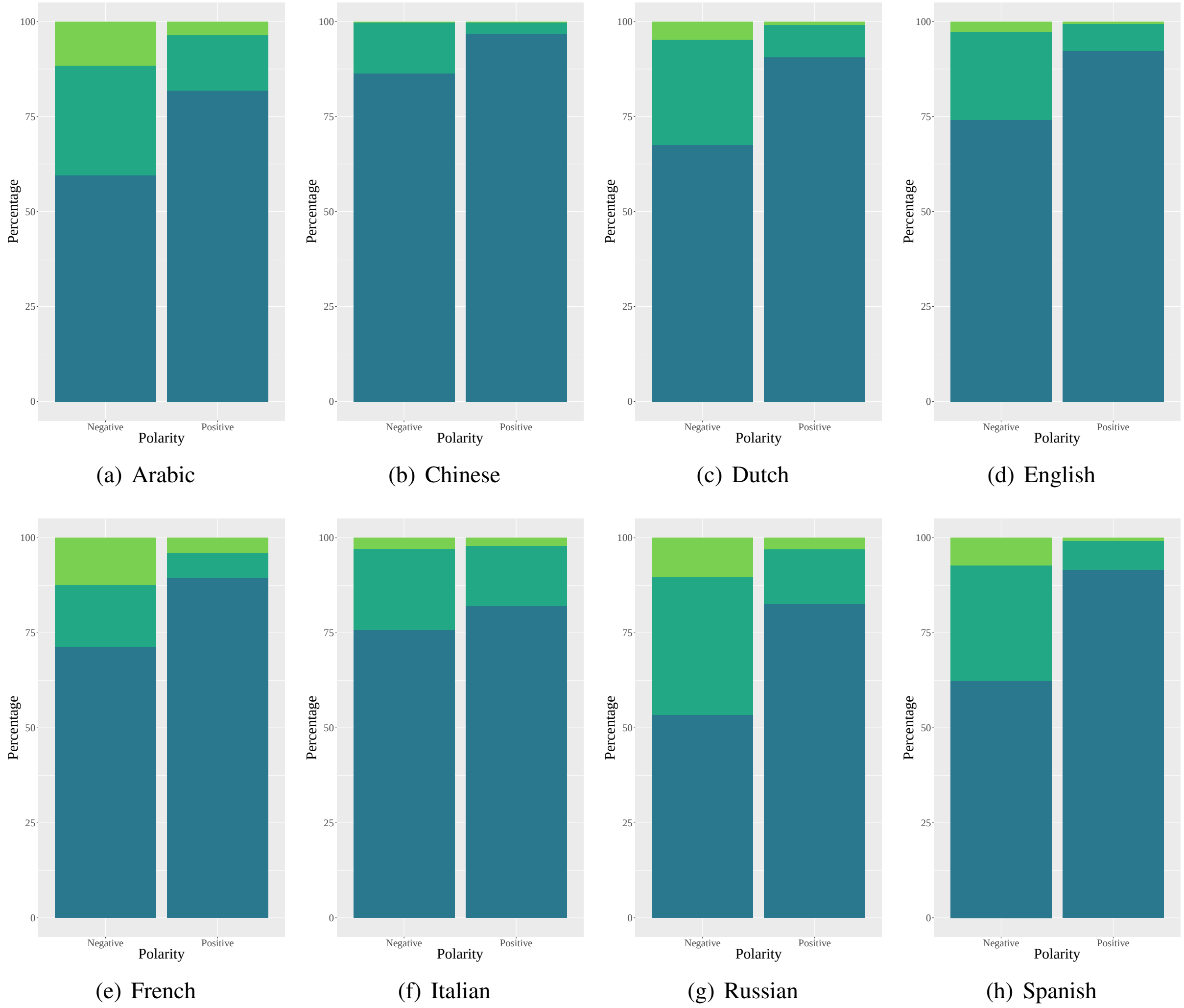
The choice of the models' hyper-parameters was based on two (contrasting) criteria: i) conservativeness with those proposed in the original models and ii) comparability among the models in this work. In particular, we ensured that each model had the same sentence vector dimensionality: 300. The only exception is SDAE: we kept the recommended value of 2400. Paragraph Vector DBOW and SkipGram were trained for 10 epochs, with a window size of 10, a minimum frequency count of 5, and a sampling threshold of 10^{-5}. FastSent was set as having a minimum count of 3, and no sampling. The probabilities in the corruption function of the SDAE were set as $p_o = 0.1$ (deletion) and $p_x = 0.1$ (swapping). The dimension of the RNN (GRU) hidden states (and hence sentence vector) was 2400, whereas single words were assigned 100 dimensions. The learning rate was set to 0.01 without decay, and the training lasted 7.2 hours on a NVIDIA Titan X GPU. The main properties of each algorithm are summarised in Table 1.

Algorithm	**WO**	**SO**
Additive SkipGram		
ParagraphVec DBOW		
FastSent		✓
Sequential Denoising AutoEncoder	✓	

Table 1: Sensitivity to Word or Sentence Order.

[2] https://github.com/fh295/SentenceRepresentation

[3] https://radimrehurek.com/gensim/models/doc2vec.html



(a) Arabic
(b) Chinese
(c) Dutch
(d) English
(e) French
(f) Italian
(g) Russian
(h) Spanish

Figure 1: Percentages of negative (left) and positive (right) sentences with the same amount of negative grammatical markers. A count of 0 is represented in dark blue, 1 in light blue, and 2 or more in green.

4 Experimental Setup

Now, we evaluate the quality of the distributed sentence representations from § 3 on Sentiment Analysis. In § 4.1 we introduce the datasets of all the considered languages, and the evaluation protocol in § 4.2. Finally, to provide a potential performance ceiling, we compare the obtained results with those of a deep, state-of-art classifier, outlined in § 4.3.

4.1 Datasets

The data for training and testing are sourced from the SemEval 2016: Task 5 (Pontiki et al., 2016). These datasets provide customer reviews in 8 languages labelled with Aspect-Based Sentiment, i.e., opinions about specific entities or attributes rather than generic stances. The languages include Arabic (hotels domain), Chinese (electronics), Dutch (restaurants and electronics), English (restaurants and electronics), French, Russian, Spanish, and Turkish (restaurants all). We mapped the labels to an overall polarity class (positive or negative) by selecting the majority class among the aspect-based sentiment classes for a given sentence. Note that no general sentiment for the sentence was included in this pool. Moreover, we added data for Italian (tweets) from the SENTIPOLC shared task in EVALITA 2016 (Barbieri et al., 2016). We discarded neutral stances from the corpus, and retained only positive and negative ones. Table 2 shows the final size of the dataset partitions and the Wikipedia dumps. In Figure 1, we report the percentage of sentences with the same amount of negative grammatical markers (e.g. the word *not* and the suffix *n't* in English) based on their polarity class. We discuss the impact of the variation of these percentages on the results in § 5.

Language	Wikipedia Dumps	Train	Test
Arabic	3406732	4570	1163
Chinese	8067971	2593	1011
Dutch	11860559	2169	683
English	30000002	3584	1102
French	26024881	1410	534
Italian	15338617	4588	512
Russian	16671224	2555	835
Spanish	22328668	1553	646
Turkish	3622336	1008	121

Table 2: Size of the data partitions (# sentences).

4.2 Evaluation Protocol

After mapping each sentence in the dataset to its distributed representation, we fed them to a Multi-Layer Perceptron (MLP), trained to detect the sentence polarity. In the MLP, a logistic regression layer is stacked onto a 60-dimensional hidden layer with a hyperbolic tangent activation. The weights were initialised from the random *xavier* distribution Glorot and Bengio (2010). The cross-entropy loss was normalised with the L2-norm of the weights scaled by $\lambda = 10^{-3}$. The optimisation with gradient descent ran for 20 epochs with early stopping. Batch size was 10 and the learning rate 10^{-2}.

4.3 Comparison with State-of-Art Models

In addition to unsupervised distributed sentence representations, we test a bi-directional Long Short-Term Memory neural network (bi-LSTM) on the same task. This is a benchmark to compare against results of deep state-of-art architectures. The choice is based on the competitive results of this algorithm and on its sensitivity to word order. The accuracy of this architecture is 45.7 for 5-class and 85.4 for 2-class Sentiment Analysis on the standard dataset of the Stanford Sentiment Treebank.

The importance of word order is evident from the architecture of the network. In a recurrent model, the word embedding of a word w_t at time t is combined with the hidden state h_{t-1} from the previous time step. The process is iterated throughout the whole sequence of words of a sentence. This model can be extended to multiple layers. LSTM is a refinement associating each time epoch with an input, control and memory gate, in order to filter out irrelevant information (Hochreiter and Schmidhuber, 1997). This model is bi-directional if it is split in two branches reading simultaneously the sentence in opposite directions (Schuster and Paliwal, 1997).

Contrary to the evaluation protocol sketched in § 4.2, the bi-LSTM does not utilise unsupervised sentence representations. Rather, it is trained directly on the datasets from § 4.1. The optimisation ran for 20 epochs, with a batch size of 20 and a learning rate of $5 \cdot 10^{-2}$. The 60-dimensional hidden layer had a dropout probability of 0.2. Crucially, the word embeddings were initialised with the Skip-Gram model described in § 3.1.1. Since performance tends to vary depending on the initialisation, this ensures a fair comparison.

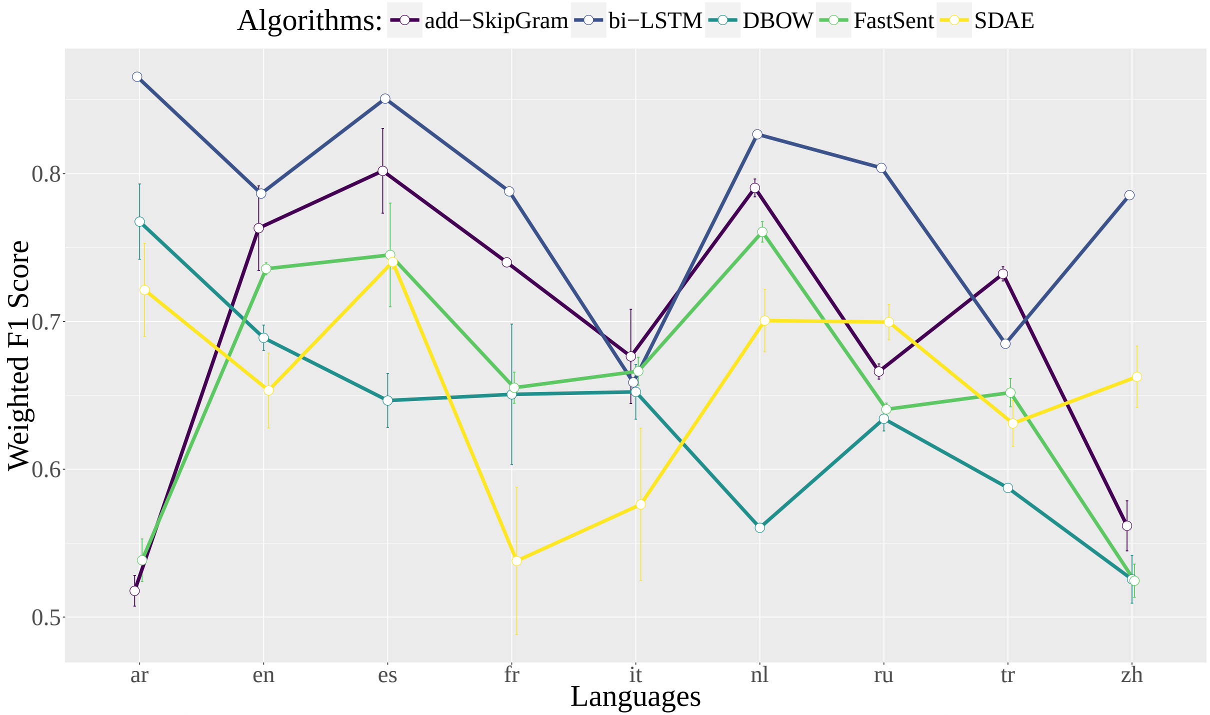
5 Results

The results are displayed in Figure 2. Weighted F1 scores were preferred over accuracy scores, since the two classes (positive and negative) are unbalanced. We decoded the unsupervised representations multiple times through different initialisation of the MLP weights, hence we report both the mean value and its standard deviation. The results are not straightforward: there is no algorithm outperforming the others in each language; unexpectedly not even the bi-LSTM used as a ceiling. However, the variation in performance follows certain trends, depending on the properties of languages and algorithms. We now examine: i) how performance is affected by the properties of the algorithms, such as those summarised in Table 1; ii) how typological features concerning negation and the text domain could make polarity harder to detect; iii) the interaction between negation and indefinite pronouns, by visualising the contribution of each word to the predicted class probabilities.

5.1 Feature Sensitivity of the Algorithms

The state-of-art bi-LSTM algorithm chosen as a ceiling is not the best choice in some languages (Italian, and Turkish). In these cases, it is always surpassed by the same model: additive Skip-Gram. The drop in Italian is possibly linked to its dataset in specific, since all the algorithms behave similarly badly. Turkish is possibly challenging for a recursive model because of the sparsity of its vocabulary. These cases, however, are not isolated: averaged word embeddings outperformed LSTMs in text similarity tasks (Arora et al., 2016) and provide a strong baseline in English (Adi et al., 2017).

In any case, the general high performance of additive Skip-Gram is noteworthy: it shows that a simple method achieves close-to-best results in almost every language among decoded distributed



	add-SkipGram	DBOW	FastSent	SDAE	bi-LSTM
ar	51.76 ± 1.78	**76.76** ± 4.42	53.85 ± 2.50	72.13 ± 5.44	86.56
en	**76.31** ± 4.96	68.89 ± 1.49	73.57 ± 0.71	65.33 ± 4.39	78.65
es	**80.19** ± 4.96	65.65 ± 3.17	74.50 ± 6.07	74.03 ± 0.81	85.08
fr	**74.00** ± 0.43	65.07 ± 8.23	65.52 ± 1.82	53.80 ± 8.63	78.80
it	**67.63** ± 5.52	65.24 ± 3.20	66.63 ± 1.63	57.62 ± 8.92	*65.88*
nl	**79.04** ± 1.05	56.04 ± 0.00	76.06 ± 1.20	70.06 ± 3.65	82.66
ru	66.61 ± 0.89	63.42 ± 1.43	64.05 ± 0.74	**69.96** ± 2.07	80.39
tr	**73.22** ± 0.84	58.74 ± 0.00	65.19 ± 1.66	63.10 ± 2.69	*68.49*
zh	56.17 ± 2.93	52.58 ± 2.80	52.46 ± 1.95	**66.26** ± 3.61	78.55

Figure 2: Results of 5 different algorithms on 9 languages. Values report the mean Weighted F1 Score and the standard deviation. The best results per language are given in bold and the second-best is underlined. Data points where the ceiling is outperformed are in italics.

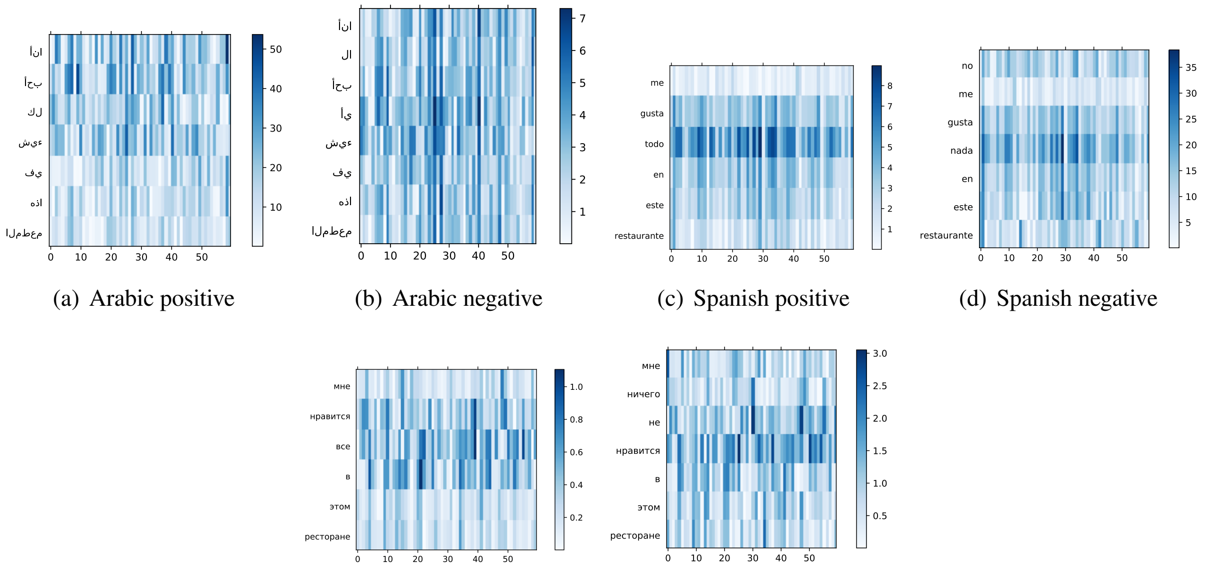
representations. This result is in line with other findings: Wieting et al. (2016b) showed that word embeddings, once retrained and decoded by linear regression, beat many methods that generate sentence representations directly.

Moreover, the second-best method for languages is always FastSent, which is the only one hinging upon neighbouring sentences as features. This demonstrate that sentiment is encoded not only within a sentence, but also in its textual context. As a consequence, a relatively small and accessible dataset (Wikipedia) is sufficient to provide a reliable model in most languages. Nonetheless, the varying size of the dumps affects FastSent as well as the other unsupervised algorithms: limited data hinders them from learning faithful representations, as in Arabic, Chinese, and Turkish (see Table 2).

In general, algorithms sensitive to the same features behave similarly, e.g. SDAE and bi-LSTM. They follow the same trend in relative improvements from one language to another. The generally low performance of SDAE could depend on the limited training time, which was necessary to evaluate the algorithm on the whole set of languages.

5.2 Typology of Negation and Domain

In some languages, the scores are very scattered: this fluctuation might be due to their peculiar morphological properties. In particular, Arabic is an introflexive language, Chinese is a radically isolating language, and Turkish an agglutinative language. On the other hand, the algorithms achieve better



(a) Arabic positive

(b) Arabic negative

(c) Spanish positive

(d) Spanish negative

(e) Russian positive

(f) Russian negative

Figure 3: Visualization of the derivative of the class scores with respect to the word embeddings.

scores in the fusional languages, save Italian.

A fine-grained analysis shows also that the performance is affected by the typology of the negation in each language, although negative markers appear in a reduced number of examples (see Figure 1). Semantically, negation is crucial in switching or mitigating the polarity of lexical items and phrases. Morpho-syntactically, negation is expressed through several constructions across the languages of the world. Constructions differ in many respects, which are classified as feature-value pairs in databases like the World Atlas of Language Structures (Dryer and Haspelmath, 2013).[4]

Negation can affect the declarative verbal main clauses. In fact, negative clauses can be: i) symmetric, i.e., identical to the affirmative counterpart except for the negative marker; ii) asymmetric, i.e. showing structural differences between negative and affirmative clauses (in constructions or paradigms); iii) showing mixed behaviour. Alterations concern for instance finiteness, the obligatory marking of unreality status, or the expression of verbal categories. Secondly, negation interacts with indefinite pronoun (e.g. *nobody*, *nowhere*, *never*). Negative indefinites can i) co-occur with standard negation; ii) be forbidden in concurrence; iii) display a mixed behaviour. Finally, the relation of the negative marker with respect to verb is prone to change. Firstly, it can be either an affix or a prosodically independent word. Secondly, its position can be anchored to the verb (preceding, following, or both). Thirdly, negation can be omitted, doubled or even tripled.

Performances seem to suffer the ambiguity in mapping between a negative marker and negative meaning. In fact, the bi-LSTM achieves lower scores in languages with asymmetric constructions (Chinese, English, and Turkish): the additional changes in the sentence construction and/or verb paradigm might create noise. Additional reasons of difficulty may occur when negation is doubled (French) or affixed (Turkish), since this makes negation redundant or sparse. On the other hand, add-SkipGram appears to be sensitive to the presence of negation: according to the counts in Figure 1, when this is too pervasive (Arabic and Russian) or rare (Chinese), the scores tend to decrease.

These comments on the results based on linguistic properties can also suggest speculative solutions for future work. For algorithms based on sentence order, it is not clear whether the problem lies in the lack of wider collections of texts in some languages, or rather on the maximum amount of information about polarity that is learnt through a sentence-level distributional hypothesis. On the other hand, impairments of the other algorithms seem to be linked

[4] The features considered here for negation are 113A 'Symmetric and Asymmetric Standard Negation', 114A 'Subtypes of Asymmetric Standard Negation', 115A 'Negative Indefinite Pronouns and Predicate Negation', and 143A 'Order of Negative Morpheme and Verb'.

with redundancies and noise. Filtering out words that contribute to this effect might benefit the quality of the representation. Moreover, the sparsity due to cases where negation is an affix might be mitigated by introducing character-level features.

The other inherent source of variation is the text domain, on which the difficulty of the task depends (Glorot et al., 2011). Although the unstructured nature of tweets could hinder the quality of the sentence representations in Italian, however, no clear effect is evident based on the other domains.

5.3 Visualisation

Since languages vary in the "polarity agreement" between verbs and indefinite pronouns, algorithms may weigh these as features differently. We analyse their role through a visualizasion of the activation in the hidden layer of the bi-LSTM. In particular, we approximated the objective function through a linear function, and estimated the contribution of each word to the true class probability by computing the prime derivative of the output scores with respect to the embeddings. This technique is presented and detailed by Li et al. (2015). The visualised hidden layers are shown in Figure 3, whereas the sentences used as input are glossed in Ex. (3) (Arabic), Ex. (4) (Spanish), and Ex. 5 (Russian).

(3) *'ana 'uhibu kl shay' fi hadha*
1SG like.NPST.1SG every thing in this
almataeim / 'ana la 'uhibu
restaurant / 1SG not.NPST like.NPST.1SG
'ayu shay' fi hadha almataeam
any thing in this restaurant

(4) *me gust-a todo en est-e*
1SG.DAT like-3SG everything in this-SG
restaurant-e / no me gust-a
restaurant-SG / not 1SG.DAT like-3SG
nada en est-e restaurant-e
nothing in this-SG restaurant-SG

(5) *mne nráv-itsja vs-jo v*
1SG.DAT like.IMPV-PRS.3SG all-NOM.SG in
ét-om restoráne / mne
this-PREP.SG restaurant-PREP.SG / 1SG.DAT
ni-čevó ne nráv-itsja v
nothing-GEN not like.IMPV-PRS.3SG in
ét-om restorán-e
this-PREP.SG restaurant-PREP.SG

The two compared sentences correspond to the translation of two English sentences. The first is positive: 'I like everything in this restaurant'; the second is negative: 'I don't like anything in this restaurant'. These include a domain-specific but sentiment-neutral word that plays the role of a touchstone. The more a cell tends to blue, the higher its activation. In some languages (e.g. Arabic), the sentiment verb elicits a stronger reaction in the positive polarity, whereas the indefinite pronoun dominates in the negative polarity. In several other languages (e.g. Spanish), indefinite pronouns are more relevant than any other feature. In Russian, only sentiment verbs always provoke a reaction. These differences might be related to the "polarity agreement" of these languages, which happens always, sometimes, and never, respectively. In some other languages, however, no evidence is found of any similar activation pattern.

6 Conclusion

In this work, we examined how much sentiment polarity information is retained by distributed representations of sentences in multiple typologically diverse languages. We generated the representations through various algorithms, sensitive to different properties from training corpora (e.g, word or sentence order). We decoded them through a simple MLP and compared their performance with one of the state-of-art algorithms for Sentiment Analysis: bi-directional LSTM. Unexpectedly, for some languages the bi-directional LSTM is outperformed by unsupervised strategies like the addition of the word embeddings obtained from a Skip-Gram model. This model, in turn, surpasses more sophisticated algorithms for most of the languages. This demonstrates i) that no algorithm is the best across the board; and ii) that some simple models are to be preferred even for downstream tasks, which partially contrasts with the conclusions of Hill et al. (2016). Moreover, representation algorithms sensitive to word order have similar trends, but they do not always achieve performance superior to algorithms based on the sentence order. Finally, some properties of languages (i.e. their type of negation) appear to have an impact on the scores: in particular, the asymmetry of negative and affirmative clauses and the doubling of negative markers.

Acknowledgements

This work was supported by the ERC Consolidator Grant LEXICAL (648909). The authors would like to thank the anonymous reviewers for their helpful suggestions and comments.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2017. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *Proceedings of ICLR*. http://arxiv.org/abs/1608.04207.

Mariana SC Almeida, Cláudia Pinto, Helena Figueira, Pedro Mendes, and André FT Martins. 2015. Aligning opinions: Cross-lingual opinion mining with dependencies. In *Proc. of the Annual Meeting of the Association for Computational Linguistics*.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2016. A simple but tough-to-beat baseline for sentence embeddings. In *ICLR 2017*.

Alexandra Balahur and Marco Turchi. 2014. Comparative experiments using supervised learning and machine translation for multilingual sentiment analysis. *Computer Speech & Language* 28(1):56–75.

Francesco Barbieri, Valerio Basile, Danilo Croce, Malvina Nissim, Nicole Novielli, and Viviana Patti. 2016. Overview of the evalita 2016 sentiment polarity classification task. In *Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016)*.

Jordan Boyd-Graber and Philip Resnik. 2010. Holistic sentiment analysis across languages: Multilingual supervised latent dirichlet allocation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 45–55.

Xilun Chen, Ben Athiwaratkun, Yu Sun, Kilian Weinberger, and Claire Cardie. 2016. Adversarial deep averaging networks for cross-lingual sentiment classification. *arXiv preprint arXiv:1606.01614* .

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *CoRR* abs/1705.02364. http://arxiv.org/abs/1705.02364.

Östen Dahl. 1979. Typology of sentence negation. *Linguistics* 17(1-2):79–106.

Kerstin Denecke. 2008. Using sentiwordnet for multilingual sentiment analysis. In *Data Engineering Workshop, 2008. ICDEW 2008. IEEE 24th International Conference on*. pages 507–512.

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig. http://wals.info/.

Alejandro Moreo Fernández, Andrea Esuli, and Fabrizio Sebastiani. 2015. Distributional correspondence indexing for cross-lingual and cross-domain sentiment classification. *Journal of Artificial Intelligence Research* 55:131–163.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*. volume 9, pages 249–256.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)*. pages 513–520.

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. *arXiv preprint arXiv:1602.03483* .

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

Mohit Iyyer, Varun Manjunatha, Jordan L Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *ACL (1)*. pages 1681–1691.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*. pages 3294–3302.

Filippos Kokkinos and Alexandros Potamianos. 2017. Structural attention neural networks for improved sentiment analysis. In *EACL 2017*.

Quoc V Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*. volume 14, pages 1188–1196.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2015. Visualizing and understanding neural models in nlp. *arXiv preprint arXiv:1506.01066* .

Bin Lu, Chenhao Tan, Claire Cardie, and Benjamin K Tsou. 2011. Joint bilingual sentiment classification with unlabeled parallel corpora. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, pages 320–330.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. *SemEval-2014* .

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. pages 3111–3119.

Dmitrijs Milajevs, Dimitri Kartsaklis, Mehrnoosh Sadrzadeh, and Matthew Purver. 2014. Evaluating neural word representations in tensor-based compositional settings. *idea* 10(47):39.

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

Amr El-Desoky Mousa and Björn Schuller. 2017. Contextual bidirectional long short-term memory recurrent neural network language models: A generative approach to sentiment analysis. In *EACL 2017*.

Tamara Polajnar, Laura Rimell, and Stephen Clark. 2015. An exploration of discourse-based sentence spaces for compositional distributional semantics. In *Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics (LSDSem)*. page 1.

Adam Poliak, Pushpendre Rastogi, M Patrick Martin, and Benjamin Van Durme. 2017. Efficient, compositional, order-sensitive n-gram embeddings. *EACL 2017* page 503.

Edoardo Maria Ponti. 2016. Divergence from syntax to linear order in ancient greek lexical networks. In *The 29th International FLAIRS Conference*.

Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeny Kotelnikov, Nuria Bel, Salud Marıa Jiménez-Zafra, , and Gülsen Eryigit. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval*. volume 16.

Laura Rimell, Jean Maillard, Tamara Polajnar, and Stephen Clark. 2016. Relpron: A relative clause evaluation dataset for compositional distributional semantics. *Computational Linguistics* .

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45(11):2673–2681.

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

Md Arafat Sultan, Steven Bethard, and Tamara Sumner. 2015. Dls@ cu: Sentence similarity from word alignment and semantic vector composition. In *Proceedings of the 9th International Workshop on Semantic Evaluation*. pages 148–153.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*. ACM, pages 1096–1103.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016a. Charagram: Embedding words and sentences via character n-grams. *arXiv preprint arXiv:1607.02789* .

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016b. Towards universal paraphrastic sentence embeddings. In *ICLR 2017*.

Ainur Yessenalina and Claire Cardie. 2011. Compositional matrix-space models for sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 172–182.

Guangyou Zhou, Tingting He, Jun Zhao, and Wensheng Wu. 2015. A subspace learning framework for cross-lingual sentiment classification with partial parallel data. In *Proceedings of the international joint conference on artificial intelligence, Buenos Aires*.

Xinjie Zhou, Xianjun Wan, and Jianguo Xiao. 2016. Cross-lingual sentiment classification with bilingual document representation learning .