ISSN 2499-4553

IJCoL

Italian Journal
of Computational Linguistics

Rivista Italiana
di Linguistica Computazionale

Volume 8, Number 2
december 2022

aA ccademia
university
press

IJCoL
Italian Journal of Computational Linguistics

8-2 | 2022

Italian Journal of Computational Linguistics vol. 8, n. 2 december 2022

Electronic version
URL: https://journals.openedition.org/ijcol/1036
DOI: 10.4000/ijcol.1036
ISSN: 2499-4553

Publisher
Accademia University Press

Electronic reference
IJCoL, 8-2 | 2022, “Italian Journal of Computational Linguistics vol. 8, n. 2 december 2022” [Online], Online since 01 December 2022, connection on 22 February 2023. URL: https://journals.openedition.org/ijcol/1036; DOI: https://doi.org/10.4000/ijcol.1036

Registrazione presso il Tribunale di Trento n. 14/16 del 6 luglio 2016

Rivista Semestrale dell'Associazione Italiana di Linguistica Computazionale (AILC)

Associazione Italiana di
Linguistica Computazionale

direttore responsabile
Michele Arnese

isbn 9791255000488

Accademia University Press
via Carlo Alberto 55
I-10123 Torino
info@aAccademia.it
www.aAccademia.it/IJCoL_8_2

Accademia University Press è un marchio registrato di proprietà
di LEXIS Compagnia Editoriale in Torino srl

IJCoL

Volume 8, Number 2
december 2022

CONTENTS

AriEmozione 2.0: Identifying Emotions in Opera Verses and Arias

Shibingfeng Zhang[*]
Saarland University

Francesco Fernicola[**]
Università di Bologna

Federico Garcea[†]
Università di Bologna

Paolo Bonora[‡]
Università di Bologna

Alberto Barrón-Cedeño[§]
Università di Bologna

We present the task of identifying the emotions conveyed by the lyrics of Italian opera arias. We shape the task as a multi-class supervised problem, considering the six emotions from Parrot's tree: love, joy, admiration, anger, sadness, and fear. We manually annotated an opera corpus with $2.5k$ *instances at the verse level and experimented with different classification models and representations to identify the expressed emotions. Our best-performing models consider character* 3*-gram representations and reach relatively low levels of macro-averaged* F_1*. Such performance reflects the difficulty of the task at hand, partially caused by the size and nature of the corpus: relatively short verses written in 18th-century Italian. Building on what we learned from the verse-level setting, we adopt a higher granularity and increase the size of the corpus. First, we switch from verses to arias in order to have longer and more expressive texts. Second, we construct a new corpus with* $40k$ *arias (*$\sim 90k$ *verses). This new dataset contains silver data, annotated by self-learning on the basis of an ensemble of binary classifiers.*

We then experiment with more sophisticated representations, by learning an embedding space and using it to train new models for the identification of emotions at the aria level, obtaining a significant performance boost.

1. Introduction

Arias are used by authors to express the emotional state of the singing character within an opera play. In 17th- and 18th-century Italian operas, characters brought on stage passions ("*affetti*") induced in their souls by the succession of events in the drama. Musicological studies use these affects as one of the interpretative keys of the work

[*] Dept. of Language Science and Technology - Universität des Saarlandes Campus, 66123 Saarbrücken, Germany. E-mail: shzh00003@stud.uni-saarland.de
[**] Dept. of Interpreting and Translation - Corso della Repubblica 136, 47121 Forlì, Italy. E-mail: francesco.fernicola2@unibo.it
[†] Dept. of Interpreting and Translation - Corso della Repubblica 136, 47121 Forlì, Italy. E-mail: federico.garcea2@unibo.it
[‡] Dept. of Classical Philology and Italian Studies - Via Zamboni 32, 40126 Bologna, Italy. E-mail: paolo.bonora@unibo.it
[§] Dept. of Interpreting and Translation - Corso della Repubblica 136, 47121 Forlì, Italy. E-mail: a.barron@unibo.it – Corresponding author

as a whole (Zoppelli 2001; McClary 2012). In AriEmozione we aim at creating models for the automatic identification of emotions in opera arias. Such models represent a valuable tool for the systematic study and organisation of the vast *repertoire* of arias and characters of this period for musicologists and the lay public alike.

Since an aria may express more than one emotion, we depart at a lower-granularity level: the verse. We first engineer models to identify the emotion of a single verse and then we point higher to identify the emotion(s) expressed by full arias. In the verse-level experiments, the small amount of data available makes it difficult to rely on dense representations or sophisticated models. A 2-layer feed-forward neural network fed with TF-IDF-weighted character 3-grams achieved the best F_1-macro of 0.47. This relatively low performance reflects the difficulty of the task at hand, partially caused by the small amount of supervised data available. In order to overcome these limitations, we produce a significantly larger annotated dataset by means of self-learning. Even if the new data is noisy, the larger amount of supervised instances allows for the application of dense representations and a convolutional neural network, resulting in a performance boost of 0.20 points absolute, passing from an F_1-macro of 0.47 to 0.67.

Our contributions can be summarised as follows.

1. We produced AriEmozione 1.0 —a manually-annotated corpus with emotion labels at the verse level including $2.5k$ instances.
2. We produced AriEmozione 2.0 —a self-learning-annotated corpus with emotion labels both at the aria and at the verse level including $40k$ arias ($90k$ verses).
3. We produced a FastText embedding space of 17th- and 18th-century Italian operas.
4. We explored supervised models for the identification of emotions in opera at the verse level.
5. We explored supervised models for the identification of emotions in opera at the aria level.

We release both corpora and the embedding space to the research community as well as the implementation of the different models both at verse and aria level.

The rest of the paper is articulated as follows. Section 2 offers some background about both opera and emotions. Section 3 reviews related work on both sentiment analysis and emotion identification. Section 4 presents the work intended to identify emotions at the verse level, including the construction of the dataset and multiple experiments. Section 5 describes the approach after switching to the aria level, including the automatic production of the corpus and the application of deep-learning models. Section 6 closes our contribution by drawing conclusions and identifying interesting research avenues for the future work.

2. Background

In music, *aria* refers to a piece of lyrics within the context of a full opera. An aria usually consists of more than one verse that composes the singer's participation in the dialogue. In general, opera lyrics are highly structured (Burden 1998); usually split in recitative parts where the action occurs, and arias where characters, normally singing a solo, express their feelings and motivations. Arias have a strophic structure, during the 17th century often dyadic with repetition of the first part (*da capo*). Some times, the first part gives a metaphoric representation of the *affetto* with the second explicating its consequences on the singing character. Each aria is conceived as a whole and as a closed piece, hence being potentially interchangeable between different plays as its function is to convey one or more distinctive *affetti* to the public.

Our research builds on top of CORAGO, the Repertoire and archive of Italian opera librettos.[1] CORAGO constitutes the first implementation of the RADAMES prototype (*Repertoriazione e Archiviazione di Documenti Attinenti al Melodramma E allo Spettacolo*; Repertorisation and Archiving of Documents Related to Melodrama And Entertainment) (Pompilio et al. 2005). All texts are written in 18th-century Italian and articulated in verses and *stanzas* —groups of verses and the way the metric and rhyme structure of a lyric is articulated. The most represented authors in our corpus are two of the most successful librettists of the 18th century: Apostolo Zeno and Pietro Metastasio whose 26 librettos were put in music in more than one thousand operas during the 19th century. Whereas Zeno composed mostly operas on historical and mythological themes, Metastasio is considered the most important writer of *opera seria*.

Most arias in the collection contain between two and three verses. We derive the emotion classification scheme from various previous works. We consider René Descartes' "Les passions de l'âme" (1649) as the reference for the coeval literature for emotions representations and their social expressions and meanings (Garavaglia 2018). We then selected a contemporary model that could be aligned in order to represent the taxonomy of Descartes while being based on the lexical representation of emotion in lyrics. We also consider Shaver et al. (1987)'s prototype approach based on the analysis of the lexicon of emotions. Through this review, carried out together with expert musicologists with extensive experience in the analysis of operas during the studied period, we settled on Parrott (2001)'s hierarchical classification and end up with six primary emotions, which turn into our six classes:

`Amore (love):` a focused sense of belonging, care and attraction toward someone; incl. affection, lust, and longing.

`Gioia (joy):` a sense of fulfillment and positiveness; incl. cheerfulness, zest, contentment, pride, optimism, enthrallment, and relief.

`Ammirazione (admiration):` admiration or adoration of someone's talent, skill, or other physical or mental qualities; incl. esteem, respect, and approval.

`Rabbia (anger):` a state of repulsion and frustration due to something or someone interfering with one's aims; incl. irritability, exasperation, rage, disgust, envy, and torment.

`Tristezza (sadness):` a state following an unwanted outcome, a loss or a delusion; incl. suffering, disappointment, shame, and neglect.

`Paura (fear):` a state induced by the interpretation of oncoming events as potentially dangerous or threatening; incl. horror and nervousness.

[1] `http://corago.unibo.it`.

During the annotation process, We included an extra class: **nessuna (none)**, which applies mostly to verses containing only non-actionable words; the few instances of this class have been neglected in all experiments (cf. Section 4).

3. Related Work

Sentiment analysis, also known as opinion mining, aims at determining the polarity of a text by investigating text features (Liu and Özsu 2009). The decision is often binary —positive vs negative— or ternary, with the addition of an intermediate neutral class. Research on sentiment analysis is vast and we refer the interested reader to Birjali et al. (2021) for a thorough overview. Starting from numerous advances on sentiment analysis, researchers attempted to move towards a finer-degree problem in the more complicated task of multi-class emotion identification. Research has been conducted on various types of text, ranging from social media contents with tweets (Roberts et al. 2012) or Facebook posts (Pool and Nissim 2016) to lyrics (Hu, Chen, and Yang 2009), news (Kirange and Deshmukh 2012), and children's fairy tales (Alm, Roth, and Sproat 2005).

Datasets exist for the analysis and identification of emotions in Italian; most of them focused on social media content. MultiEmotion-It is a corpus with comments from YouTube and Facebook posts responding to music videos and advertisement (Sprugnoli 2020). These comments are annotated according to four aspects: *relatedness*, *polarity*, *emotions* and *sarcasm*. In the specific case of emotions, Plutchik (1980)'s model is used, resulting in classes *joy*, *sadness*, *fear*, *anger*, *trust*, *disgust*, *surprise*, and *anticipation*. MultiEmotions-It (Sprugnoli 2020) and AriEmozione 1.0 (Fernicola et al. 2020) were both released in 2020, representing two of the first manually-annotated corpora for the identification of emotions in Italian. FEEL-IT is a corpus with $2k$ Italian tweets annotated with one label out of *anger*, *fear*, *joy*, and *sadness* (Bianchi, Nozza, and Hovy 2021). The justification in the selection of these labels relies on their "frequent occurrence in text" (Bianchi, Nozza, and Hovy 2021, p. 76)

Our contribution in terms of corpora release go beyond both MultiEmotion-It and FEEL-IT. Similarly to the former, we base our label selection on a formal classification of emotions supported on a psychological and philosophical theory. Similarly to the latter, instead, we narrow such selection by an expert analysis of the emotions that are more present in the analysed genre. Both MultiEmotion-It and FEEL-IT contain annotations at the document level (be it a tweet or a comment). As Strapparava et al. (2012), who released a corpus of popular music in English, we go at the sub-document level and annotate single verses and arias.

Regarding models, some approaches involve rule-based systems. For example, Asghar et al. (2017) proposed a rule-based framework for sentence-level emotion identification of user reviews using an emotion lexicon. Researchers created a mixed-mode classifier that takes into account not only emotion words, but also emoticons and slang and compared the performance of a mixed-mode classifier with another classifier that is created using only emotion words as resources. Both models are tested on a corpus of news texts and the mixed-mode classifier was the one which performed better.

Some researchers opted for a hybrid approach, making use of both supervised and unsupervised methods to achieve a higher accuracy. This is the case of Gievska et al. (2014), who designed an emotion detection approach to deal with the ISEAR dataset.[2] This study considered seven emotions derived from Ekman's six emotional categories (Ekman 1994): *anger*, *fear*, *sadness*, *disgust*, *joy*, *surprise*, and an additional *neutral* category in order to reduce the effect of misclassified data. They experimented with a lexical-based method alone, the machine learning method alone, and the blended method using both of the previous models. The lexical-based method is developed using a variety of language resources such as WordNetAffect (Strapparava and Valitutti 2004), AFINN (Nielsen 2011), H4Lvd,[3] and the NRC word-emotion association lexicon (Mohammad and Turney 2013). An SVM obtained a significant precision advantage and the hybrid method performed the best.

The existing supervised text emotion analysis research can also be categorised into three general classes: single-label learning (SLL), multi-label learning (MLL), and label distribution learning (LDL) (Zhao and Ma 2019). In SLL, the emotion of a text is represented by a single emotion such as *joyful* or *sad*. In MLL a text is assumed to transmit more than one emotion and can be therefore assigned more than one label. For example, Ye et al. (2012) constructed a reader emotion corpus collecting news articles from the Sina news media. Each news article corresponds to one to three emotion labels. Various feature selection strategies such as document frequency and chi-square statistic are tested with different multi-class classification models. LDL goes a step further and assigns not only a set of emotion labels but also the corresponding emotion intensity. Zhou et al. (2016) proposed a distribution learning approach capable of identifying emotions with their respective intensities of the given sentence. Eight emotion labels are established based on Plutchik's wheel of emotions (Plutchik 1980). Each sentence may express one or more emotions and the sum of emotion intensities for each sentence is normalised to one. This study also captures the relations among the eight emotions of Plutchik's wheel (Plutchik 1980) and incorporate these relations into the learning algorithm in order to enhance the accuracy.

4. Emotion Identification at the Verse level

This section presents our efforts to identify emotion in opera lyrics at the verse level. We cover the creation of the AriEmozione 1.0 corpus as well as the exploration of diverse models and representations.

4.1 The AriEmozione 1.0 Corpus

This corpus is a subset of the materials from project CORAGO (cf. Section 2). We selected a set of 678 operas composed between 1655 and 1765, considering only the lyrical text in the arias (and neglecting, for instance, recitatives). For the annotation, we split all opera arias into verses, resulting in 2,473 instances. At this stage, we opted for verses because we observed that the snippets hardly express more than one emotion at this level of granularity. Two native speakers of Italian annotated all verses independently following the instructions displayed in Figure 1. They were asked to include (*i*) the emotion transmitted by the verse, (*ii*) an optional secondary label (in case they perceived a second emotion), and (*iii*) their level of confidence: total confidence, partial confidence, or doubtful. Cohen's kappa inter-annotator agreement (Fleiss, Cohen, and Everitt 1969) on the primary emotion was of 0.323, which is considered as a fair agreement —this

[2] https://github.com/sinmaniphel/py_isear_dataset
[3] http://www.wjh.harvard.edu/~inquirer/Home.html.

First of all, thank you for helping with this work. We are a group of researchers from the D. of Classical Philology and Italian Studies and the D. of Interpreting and Translation, both at UniBO. Your work will help us to produce artificial intelligence models to analyse the lyrics in music.
At this stage we are focused on opera. You will annotate arie in Italian from diverse periods, looking for the emotions that they express. Your work consists of identifying the emotion expressed in each of the verses composing an aria. You can choose among six emotions (or none of them), which are defined next: [. . .]

Each row is divided in six columns:
id A unique id, tied to the verse. Do not modify it.
verse A verse, inside of an aria. This is the text that you are going to analyse.
emotion Here you can select the expressed emotion (or none of them)
emotion sec. This is available to choose a secondary emotion, in case it is really difficult to choose just one
confidence Not being 100% sure is ok. If that is the case, please let us know by choosing the right confidence level (default: "I am sure").
comments Feel free to tell us something about this instance, if you feel like.

Figure 1
Instructions given to the annotators of the emotions at the verse level in the AriEmozione 1.0 corpus.

Table 1
AriEmozione 1.0 corpus statistics per partition and class.

	amore	gioia	ammirazione	rabbia	tristezza	paura	total
train	289	274	289	414	503	166	**1,973**
dev	36	31	23	84	61	12	**250**
test	37	39	30	64	54	15	**250**
all	362	344	342	562	618	193	**2,473**

value results from the perfect matching between the two annotators in 44% of the instances. When considering the secondary emotion as well, the two annotators were in agreement on 68% of the instances. These numbers reflect the complexity of the task. The same annotators gathered together to discuss and consolidate all dubious instances and produce a consolidated label.

Table 1 shows statistics on the number of instances per class for each corpus partition. The most represented emotions are *tristezza* (sadness) followed by *rabbia* (anger): 25% and 23% of the instances, respectively. The least represented emotion is *paura*, which negatively impacted its prediction results; cf. Section 4.3). A total of 52 verses did not express any emotion and were neglected from the experiments. The average length of the verses is of 72.5 ± 31.6 characters and the corpus contains $34,608$ tokens and $4,458$ types.[4] Appendix A shows the distribution of these classes across time periods. Table 2 shows examples of verses in the corpus, including one for each of the six emotions.

4.2 Models and Representations at the Verse Level

The nature of the corpus —a small amount of short verses written in 18th-century Italian— led us to select a humble set of models and representation alternatives. The

[4] The AriEmozione 1.0 corpus is available for download at `https://zenodo.org/record/4022318`.

Table 2
Instances from the AriEmozione 1.0 corpus, including their English translation, class, and unique identifier. We include free (unofficial) translations for clarity.

verse	class (id)
Non ho più lagrime; non ho più voce; non posso piangere; non so parlar I have no more tears; I have no more voice; I cannot cry; I don't know how to speak	Tristezza (ZAP1593570_03)
Barbaro! Oh dio mi vedi divisa dal mio ben; barbaro, e non concedi ch'io ne dimandi almen You Barbarian! Oh Lord, you see me separated from my very precious; barbarian, you won't even allow me a question	Rabbia (ZAP1596431_00)
Guardami e tutto obblio e a vendicarti io volo; di quello sguardo solo io mi ricorderò Look at me, all else is forgotten and I haste to avenge you; only I shall remember that gaze	Amore (ZAP1593766_01)
Su la pendice alpina dura la quercia antica e la stagion nemica per lei fatal non è; Up on the slope of the mountain the ancient oak tree still lives on, and the adverse season poses no fatal threat	Ammirazione (ZAP1594229_00)
In questa selva oscura entrai poc'anzi ardito; or nel cammin smarrito timido errando io vo I entered this dark forest not too long ago, boldly; having now lost the path I wander around, shyly	Paura (ZAP1596807_00)
Vede alfin l'amate sponde, vede il porto, e conforto prende allor di riposar Finally, the beloved shores, the harbor, are all in sight and with them come solace and sleep	Gioia (ZAP1599979_01)

baseline is a k–Nearest Neighbors algorithm (kNN), considered due to its simplicity and success in small classification tasks (Zhang and Zhou 2007). We also experiment with multi-class SVMs, logistic regression, and neural networks. Regarding the latter, we experiment with a number of architectures with two and three hidden layers. Finally, we experiment with a FastText classifier (Joulin et al. 2017). Table 3 summarises the configurations explored.[5]

As for the text representations, we consider TF–IDF vectors of both character 3-grams and word 1-grams (no higher values of n are considered due to the size of the corpus). For pre-processing, we employ the spaCy Italian tokenizer[6] and casefold the texts. We also explore with dense representations, derived from the TF–IDF vectors, by means of both LDA (Hoffman, Bach, and Blei 2010) and LSA (Halko, Martinsson, and Tropp 2011). In both cases, we target reductions to 16, 32, and 64 dimensions. As embeddings, we adopted the pre-trained 300-dimensional Italian vectors of FastText (Joulin et al. 2017), and tried with character 3-grams and words.

[5] The code is available at https://github.com/TinfFoil/AriEmozione. We used Sklearn for the kNN, SVM, and logistic regression models; Keras for the neural networks, and the Facebook-provided library for FastText (cf. https://scikit-learn.org, https://keras.io and https://github.com/facebookresearch/fastText).

[6] https://spacy.io/models/it

Table 3
Experimental settings for the emotion identification models at the verse level.

Model	Settings
k-NN	L2-Norm exploring with $k \in [1, \ldots 9]$.
SVM	RBF exploring with $c \in [1, 10, 100, 1000]$ and $\gamma \in [1e-3, 1e-4]$.
Log Reg	Multinomial Logistic Regression with Newton-CG solver.
NN	2 (3) hidden layers with size $\in [32, 64, 96, 128, 256]$ ($\in [8, 16, 32, 64, 96]$); 20% dropout; ReLu for input/hidden layers; softmax for output layer; categorical cross-entropy loss function; Adam optimiser; epochs $\in [1, \ldots 15]$
FastText	300d embeddings with or without pre-training; learning rate $\in [0.3, 0.6, 1]$; epochs $\in [1, 3, 5, 10, \ldots, 100]$

4.3 Experiments at the Verse Level

We conducted several experiments to find the best combination of parameters and representations. Given the amount of instances available, we merged the training and development partitions and performed 10-fold cross validation. As standard, the test partition was left aside and only one prediction was carried out on it, after identifying the best configurations. We evaluate our models on the basis of accuracy and weighted macro-averaged F_1 to account for the class imbalance. Table 4 shows the results obtained with some interesting configurations and representations both for the cross-validation and on the test set.[7] TF-IDF character and word n-grams, LSA, and LDA were tested with all models except for FastText, on which we test with and without pre-trained embeddings. Notice that we are not interested in combining features, but in observing their performance in isolation.

The most promising representation on cross-validation is the simple character 3-grams, with which we obtained the best results across all models; although it also features the highest variability across folds. Among all 3-gram derived representations, LDA consistently obtained the worst results across all models. Still, it is more stable across folds than the sparse 3-gram representation. LSA performs significantly better than LDA and is always close to the TF-IDF words representation, most notably using the k-NN model. As for FastText, with the same epoch number and learning rate, the character 3-gram vectors always achieved much higher accuracy than the word vectors. Similar patterns are observed when predicting on the unseen test set. The character 3-grams in general hold the best performance, while the 3-gram LDA tends to remain the worst in spite of the model used. This behavior does not hold in all cases. For instance, the logistic regression model achieves F_1 =0.44 on cross-validation, but drops to 0.42 on test. This might be the result of over-fitting.

Table 5 shows the confusion matrix for the best model on test. All models tend to mix `rabbia` and `tristezza`. These two emotions get confused with each other on an average of 18% of the cases. The classifiers tend to confuse `ammirazione` for `gioia` as well, which is understandable given their semantic closeness.

A number of factors contribute to the relatively low performance. First, the verses tend to be very short, causing the identification of emotions difficult. The ancient nature

[7] The full batch of results is available at `https://docs.google.com/spreadsheets/d/1Ztjry2mJs6ufCZM1O5CQRyZ8pA5YDnToN0h0NGX1nW0/edit?usp=sharing`

Table 4
F_1 and accuracy for the emotion identification at the verse level on cross-validation and held-out test for some of the model and representation combinations.

model	representation	10-fold CV		test	
		F_1	Acc	F_1	Acc
k–NN	char 3-grams	0.38	38.51	0.35	35.15
	words	0.36	36.08	0.35	34.73
	LSA char	0.36	35.26	0.33	32.64
	LDA char	0.30	29.97	0.31	30.54
SVM–RBF	char 3-grams	0.44	43.70	0.43	43.00
	words	0.42	42.00	0.44	44.00
	LSA char	0.39	39.00	0.40	40.00
	LDA char	0.28	28.00	0.30	30.00
Log reg	char 3-grams	0.44	**45.57**	0.42	43.10
	words	0.41	43.20	0.41	43.10
	LSA char	0.36	36.30	0.34	34.73
	LDA char	0.28	30.63	0.29	30.96
2-layers NN	char 3-grams	0.42	43.61	**0.47**	**46.86**
	words	0.42	42.91	0.43	43.10
	LSA char	0.35	35.63	0.36	37.24
	LDA char	0.27	29.56	0.27	31.80
3-layers NN	char 3-grams	**0.49**	41.86	0.40	41.84
	words	0.47	42.60	0.40	41.84
	LSA char	0.44	41.86	0.41	41.84
	LDA char	0.26	31.41	0.30	31.80
FastText	char 3-grams	0.43	45.00	0.41	42.37
	pre-trained char 3-grams	0.43	47.00	0.41	41.00
	words	0.42	42.56	0.39	44.07
	pre-trained words	0.38	41.00	0.40	42.00

of the language causes pre-trained vectors, such as FastText's, to have a low word coverage. Last, but not least, the number of instances available for training is fairly small. We address these issues in the next section, where we also jump from the verse- to the aria-level emotion identification.

5. Emotion Identification at the Aria Level

We address the issues observed while experimenting at the verse level in different ways. Among them, we expand the size of the supervised data and shift to a higher granularity: the aria. This shift is motivated by the complex structure of the texts, where the lexicon and phrases used to express an *affetto* often span beyond a single verse and even a whole *stanza*. The creation of more annotated instances also opens the door to produce more sophisticated representations; e.g., in-domain embedding spaces. We open the discussion with the creation of the AriEmozione 2.0 corpus and continue exploring with diverse models and representations.

Table 5
Confusion matrix for the 2-layers neural network with TF-IDF character 3-grams on the verse-level prediction task.

	ammirazione	amore	gioia	paura	rabbia	tristezza
ammirazione	**0.37**	0.03	0.18	0.07	0.11	0.06
amore	0.03	**0.43**	0.13	0.00	0.09	0.17
gioia	0.27	0.16	**0.31**	0.20	0.09	0.07
paura	0.10	0.03	0.00	**0.40**	0.02	0.07
rabbia	0.20	0.14	0.03	0.13	**0.64**	0.17
tristezza	0.17	0.14	0.13	0.07	0.19	**0.48**

5.1 The AriEmozione 2.0 Corpus

As observed in Section 4.1, the CORAGO-1700 corpus is composed of Italian operas and lacks any supervision; the annotated AriEmozione 1.0 represents just a tiny subset. We produced corpus AriEmozione 2.0 by performing a self-learning annotation process (Jurkiewicz et al. 2020) on another subset of CORAGO-1700. The first step to label this new corpus would be to automatically identify the class of the new verses with some of our existing models and iteratively add fresh instances to the training set. Nevertheless, even the best-performing multi-class model trained on AriEmozione 1.0 achieves an F_1-measure lower than 0.47 (cf. the 2-layers NN with TF-IDF character 3-grams in Section 4.3). Hence, we adopt a one-versus-all approach (OVA) (Aly 2005). OVA decomposes the k-class classification into k binary classification problems to focus on one emotion class at a time. The instance labels are determined by the class that obtained the maximum classification score. We run parallel processes considering all six classes to iteratively produce the annotations, which end up as the silver data in the AriEmozione 2.0 corpus. Appendix B describes the process in detail.[8]

In order to assess the quality of this pre-selection, we evaluated three different binary models for each class; each model differs with regards to the training material they have access to: (*i*) AE1_{tr} is trained on the manually-annotated instances from AriEmozione 1.0 , (*ii*) $\text{AE1}_{tr}\cup\text{RAW}_{pos}$ considers all training material from AriEmozione 1.0 plus only the instances that have been assigned the class of the corresponding emotion, and (*iii*) $\text{AE1}_{tr}\cup\text{RAW}_{all}$ considers all training material from AriEmozione 1.0 plus all new instances, regardless of the class they were labelled with.

Table 6 shows the results on the binary settings over the development set of AriEmozione 1.0 . In terms of precision of the positive (emotion) class, $\text{AE1}_{tr}\cup\text{RAW}_{pos}$ performs consistently the best. Except for emotions *ammirazione* and *tristezza*, $\text{AE1}_{tr}\cup\text{RAW}_{all}$ achieves both the highest accuracy and F_1 scores. However, as precision of the emotion class is the most important metric, we adopt the strategy where only new instances predicted as belonging to the corresponding emotion are integrated.

[8] The AriEmozione 2.0 corpus is available at `https://zenodo.org/record/7097913`.

Table 6
Per-class binary evaluation of the models considering different partitions of the self-training pre-labelled instances from the CORAGO corpus. We include the precision of the positive (emotion) class. AE1$_{tr}$=binarised AriEmozione 1.0 training set; RAW$_{all}$=all new instances, regardless of their assigned label; RAW$_{pos}$=new instances labeled as (emotion) positive class.

Emotion	Training material	Acc	F$_1$	Precision
ammirazione	AE1$_{tr}$	0.851	0.787	0.019
	AE1$_{tr}$∪RAW$_{all}$	0.881	0.872	0.455
	AE1$_{tr}$∪RAW$_{pos}$	**0.882**	**0.878**	**0.530**
amore	AE1$_{tr}$	0.861	0.800	0.024
	AE1$_{tr}$∪RAW$_{all}$	**0.886**	**0.889**	0.671
	AE1$_{tr}$∪RAW$_{pos}$	0.857	0.867	**0.696**
gioia	AE1$_{tr}$	0.853	0.808	0.111
	AE1$_{tr}$∪RAW$_{all}$	**0.866**	**0.856**	0.407
	AE1$_{tr}$∪RAW$_{pos}$	0.847	0.848	**0.504**
paura	AE1$_{tr}$	0.921	0.899	0.111
	AE1$_{tr}$∪RAW$_{all}$	**0.968**	**0.970**	0.917
	AE1$_{tr}$∪RAW$_{pos}$	0.952	0.956	**0.924**
rabbia	AE1$_{tr}$	0.789	0.749	0.241
	AE1$_{tr}$∪RAW$_{all}$	**0.812**	**0.810**	0.586
	AE1$_{tr}$∪RAW$_{pos}$	0.802	0.802	**0.589**
tristezza	AE1$_{tr}$	0.746	0.724	0.296
	AE1$_{tr}$∪RAW$_{all}$	**0.782**	**0.779**	0.516
	AE1$_{tr}$∪RAW$_{pos}$	0.747	0.751	**0.611**

Table 7
Class statistics at the verse level for AriEmozione 2.0 .

	amore	gioia	ammirazione	rabbia	tristezza	paura	total
freq.	13,363	13,226	13,915	17,587	25,499	6,359	**89,949**

To produce the actual annotations that will turn into the AriEmozione 2.0 corpus, we train six new binary neural networks with softmax output layers, each responsible for one one emotion. Each network is trained on the training plus the development sets from AriEmozione 1.0 plus the pre-selected instances belonging to the corresponding emotion class from the previous process. The consolidated —and final— label for each of the new raw instances is the one with the highest score among the six models. This approach to consolidate the labels is inspired by multi-class settings such as multi-class SVMs, where the decision is based on a winner-takes-all strategy (Duan and Keerthi 2005; Crammer and Singer 2001). Table 7 shows the class distribution of the AriEmozione 2.0 corpus. It contains $90k$ verses, a significantly larger amount than its predecessor. Appendix C shows the impact of these new materials on the verse-level identification task.

One of the drawbacks for the models is that the verses tend to be too short. In the rest of the paper, we shift the granularity of our instances from the verse to the aria level. Since an aria is in general composed by more than one verse, and such verses could have

Table 8
Class distribution at the aria level for AriEmozione **1.0 (gold)** —manually annotated— and AriEmozione **2.0 (silver)** —automatically annotated.

ammirazione	amore	gioia	paura	rabbia	tristezza	**1.0 (gold)**	**2.0 (silver)**
One-class instances							
■						109	2,172
	■					122	2,084
		■				107	2,099
			■			57	757
				■		194	3,224
					■	185	5,399
Two-class instances							
■	■					9	1,317
■		■				30	1,769
■			■			9	546
■				■		12	1,878
■					■	14	2,245
	■	■				13	1,407
	■		■			5	404
	■			■		7	1,456
	■				■	24	2,689
		■	■			5	642
		■		■		8	1,381
		■			■	11	2,189
			■	■		5	579
			■		■	22	1,024
				■	■	47	3,119
Overall							
■	■	■	■	■	■	995	38,380

been identified as expressing different emotions, we establish that an aria can belong to up to two emotions. The emotion of an aria is determined by the most frequent emotion label among its verses. In case of draw, the aria keeps the top-two classes.[9] In order to avoid confusion, in the following we refer to the arias derived from AriEmozione 1.0 as "gold instances", whereas those from AriEmozione 2.0 are "silver instances". Table 8 shows the statistics of the resulting dataset.

[9] If the draw involves more than two emotions, the instance is considered too noisy and it is discarded. As a result, six arias from AriEmozione 1.0 and 1,623 arias from AriEmozione 2.0 get discarded.

Table 9
Results of the emotion identification task at the aria level for a CNN with different learning rates and number of epochs. The text representation is 300-dimensional pre-trained embeddings on character 3-grams.

learning rate	epochs	accuracy	F_1
0.0001	10	0.491	0.736
0.0001	15	0.628	0.785
0.0001	20	0.614	0.789
0.001	10	0.635	0.795
0.001	15	**0.706**	**0.829**
0.001	20	0.652	0.812

5.2 Models and Representations at the Aria Level

One of the obstacles when dealing with this kind of material is its language: 18th-century Italian. This makes ineffective representing the instances with out-of-the-box pre-trained embeddings, which are built on modern text. To address this issue, we build 300-dimensional embeddings using FastText (Bojanowski et al. 2017) using both AriEmozione 1.0 and AriEmozione 2.0 as unsupervised training material. We produced character 3-gram embeddings by training during 5 epochs with a learning rate of 0.05.

As for the classification models, we opt for a multi-label setting to predict up to two classes per instance. We use a CNN with one convolutional layer (ReLU activation functions and a stride of 3), two hidden layers and the output layer. Both hidden layers have 2,500 neurons, dropout of 0.1 and sigmoid activation functions. The output layer has a six-units sigmoid function. We use binary cross-entropy and the Adam optimizer. The classification threshold is set at 0.5.[10]

5.3 Experiments at the Aria Level

The CNNs are trained on all arias in AriEmozione 2.0 (silver instances) and tested on all arias in AriEmozione 1.0 (gold instances). Table 9 shows the results after training during different epochs and with two learning rates. The best performance is obtained when training for 15 epochs with a learning rate of 0.001: $F_1 = 0.829$. Even if this score is not directly comparable to the numbers in Table 4 (different data partitions, different granularity), the allocation of more training data and the aria granularity clearly allow for much better figures.

Table 10 shows the confusion matrices of such model. Having a multi-label setting, we opt for dissecting into six matrices: one emotion against the rest. Instances of class *tristezza* are identified the best, with a precision of 0.921, whereas instances of *paura* are the most difficult, with a precision of 0.825. These outcomes can be attributed to the imbalanced distribution of instances with double labels in gold instances and silver instances. Table 8 shows that about 55% of the silver instances have two labels, while

[10] The implementation code is available at `https://github.com/TinfFoil/AriEmozione-2.0`. We used Sklearn for label encoding, Keras for the neural networks, and the Facebook-provided library for FastText (cf. `https://scikit-learn.org`, `https://keras.io` and `https://github.com/facebookresearch/fastText`).

Table 10
Normalised confusion matrices for the emotion identification task at the aria level zoomed into each of the six classes against the rest. Absolute values shown in parenthesis.

	rest	**ammirazione**
rest	0.889 (722)	0.111 (90)
ammirazione	0.104 (19)	0.896 (164)

	rest	**amore**
rest	0.948 (773)	0.052 (42)
amore	0.100 (18)	0.900 (162)

	rest	**gioia**
rest	0.965 (792)	0.035 (29)
gioia	0.161 (28)	0.839 (146)

	rest	**paura**
rest	0.983 (877)	0.017 (15)
paura	0.175 (18)	0.825 (85)

	rest	**rabbia**
rest	0.939 (678)	0.061 (44)
rabbia	0.128 (35)	0.872 (238)

	rest	**tristezza**
rest	0.893 (618)	0.107 (74)
tristezza	0.079 (24)	0.921 (279)

only 17% of the gold instances do. Many single-label instances in AriEmozione 1.0 are assigned two (or even three) labels. The best-performing model assigned three labels to 14 arias and two labels to 345 in the test set, whereas in reality only 222 arias have two labels associated.

Overall, the performance is good considering the difficulty of the task. However, there is room for improvement: the model shows robustness in the identification of each singleton emotion, but it struggles with multi-label classification.

6. Conclusions

We addressed the novel problem of identifying the emotions expressed by opera aria lyrics. This is an interesting problem because it opens the door to the creation of search engines and to the assisted organisation and curation of repertoires —both based on emotion. It is challenging because there is a lack of supervised (and unsupervised) data in the domain, and its language —17th- and 18th-century Italian— makes the use of modern semantic representations non straightforward.

We address the problem at two granularity levels: the verse and the aria. For the former, we annotated a small collection of verses with six emotions and performed numerous experiments with different models (e.g., support vector machines, logistic regression, and neural networks) and representations (e.g., character and word n-grams and word embeddings). Our results showed that neither the amount of supervised data nor the representations were enough. We then applied a self-learning approach to produce silver data to train on, produced an embedding representation out of a large-collection of non-supervised operas, and shifted to the aria granularity level, within a multi-label setting in which each instance could express up to two emotions. These efforts enabled us to try convolutional neural networks on better representations, which resulted in a large performance boost, bringing the approach closer to be applied in a practical setting.

The work on emotion identification in opera (and other kind of musical arts) can be further refined. For instance, rather than a multi-label setting, the emotion of an aria could be judged on the basis of a distribution, which considers that each item might have non-zero intensities for every single emotion (Zhao and Ma 2019). Another interesting avenue would be considering multi-modal aspects. That is, not only the written verses but also music sheets. The parallel corpus from Strapparava et al. (2012), which includes annotations on the notes and lyrics of popular music in English, can be leveraged to investigate the cooperation between textual features and musical features for emotion identification (Mihalcea and Strapparava 2012). In the case of operas, even scene representations could be taken into consideration in the decision process. Videos could play that role for popular music.

Acknowledgments

This research was carried out in the framework of CRICC: *Centro di Ricerca per l'interazione con le Industrie Culturali e Creative dell'Università di Bologna*; a POR-FESR 2014-2020 Regione Emilia-Romagna project (`https://site.unibo.it/cricc`).
We thank Ilaria Gozzi and Marco Schillaci, students at Università di Bologna, for their support in the manual annotation of the AriEmozione 1.0 corpus.

References

Alm, Cecilia Ovesdotter, Dan Roth, and Richard Sproat. 2005. Emotions from text: Machine learning for text-based emotion prediction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 579–586, Vancouver, British Columbia, Canada, October. Association for Computational Linguistics.

Aly, Mohamed. 2005. Survey on multiclass classification methods. *Technical Report*, 19:1–9.

Asghar, Muhammad Zubair, Aurangzeb Khan, Afsana Bibi, Fazal Masud Kundi, and Hussain Ahmad. 2017. Sentence-level emotion detection framework using rule-based classification. *Cognitive Computation*, 9(6):868–894.

Bianchi, Federico, Debora Nozza, and Dirk Hovy. 2021. FEEL-IT: Emotion and sentiment classification for the Italian language. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 76–83, Online, April. Association for Computational Linguistics.

Birjali, Marouane, Mohammed Kasri, and Abderrahim Beni-Hssane. 2021. A comprehensive survey on sentiment analysis: Approaches, challenges and trends. *Knowledge-Based Systems*, 226:107–134.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Burden, Michael. 1998. The new grove dictionary of opera, ed. Stanley Sadie. *Early Music*, 26(4):669–670.

Crammer, Koby and Yoram Singer. 2001. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, December.

Duan, Kai-Bo and S. Sathiya Keerthi. 2005. Which is the best multiclass svm method? an empirical study. In Nikunj C. Oza, Robi Polikar, Josef Kittler, and Fabio Roli, editors, *Multiple Classifier Systems*, pages 278–285, Seaside, CA, USA, June. Springer Berlin Heidelberg.

Ekman, Paul. 1994. All emotions are basic. *The nature of emotion: Fundamental questions*, pages 15–19. Oxford University Press.

Fernicola, Francesco, Shibingfeng Zhang, Federico Garcea, Paolo Bonora, and Alberto Barrón-Cedeño. 2020. Ariemozione: Identifying emotions in opera verses. In *Proceedings of the Seventh Italian Conference on Computational Linguistics (CLiC-it 2020)*, Bologna, Italy [online], March, 2021.

Fleiss, Joseph L., Jacob Cohen, and B.S. Everitt. 1969. Large sample standard errors of kappa and weighted kappa. *Psychological Bulletin*, 72(5):323–327.

Garavaglia, Andrea. 2018. Funzioni espressive dell'aria a metà seicento secondo il "Giasone" di Cicognini e Cavalli. *Il Saggiatore Musicale*, Anno XXV(1):5–31.

Gievska, Sonja, Kiril Koroveshovski, and Tatjana Chavdarova. 2014. A hybrid approach for emotion detection in support of affective interaction. In *2014 IEEE International Conference on Data Mining Workshop*, pages 352–359, Shenzhen, China, December. IEEE.

Halko, Nathan, Per-Gunnar Martinsson, and Joel A. Tropp. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288.

Hoffman, Matthew, Francis Bach, and David Blei. 2010. Online learning for Latent Dirichlet Allocation. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc.

Hu, Yajie, Xiaoou Chen, and Deshun Yang. 2009. Lyric-based song emotion detection with affective lexicon and fuzzy clustering method. In *International Society for Music Information Retrieval*, pages 123–128, Kobe, Japan, October.

Joulin, Armand, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain, April. Association for Computational Linguistics.

Jurkiewicz, Dawid, Łukasz Borchmann, Izabela Kosmala, and Filip Graliński. 2020. ApplicaAI at SemEval-2020 task 11: On RoBERTa-CRF, span CLS and whether self-training helps them. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1415–1424, Barcelona (online), December. International Committee for Computational Linguistics.

Kirange, Dnyaneshwar and Ratnadeep Deshmukh. 2012. Emotion classification of news headlines using SVM. *Asian Journal of Computer Science and Information Technology*, 5(2):104–106.

Liu, Ling and M. Tamer Özsu. 2009. *Encyclopedia of database systems*, volume 6. Springer New York, NY, USA:.

McClary, Susan. 2012. *Desire and Pleasure in Seventeenth-Century Music*. University of California Press, Berkeley, CA, 1 edition.

Mihalcea, Rada and Carlo Strapparava. 2012. Lyrics, music, and emotions. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 590–599, Jeju Island, Korea, July. Association for Computational Linguistics.

Mohammad, Saif M. and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

Nielsen, Finn Årup. 2011. A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. In *Proceedings of the ESWC2011 Workshop on 'Making Sense of Microposts': Big things come in small packages*, pages 93–98, Heraklion, Crete, May.

Parrott, W. Gerrod. 2001. *Emotions in Social Psychology: Essential Readings*. Psychology Press.

Plutchik, Robert. 1980. A general psychoevolutionary theory of emotion. *Theories of emotion*, 1:3–31.

Pompilio, Angelo, Lorenzo Bianconi, Fabio Regazzi, and Paolo Bonora. 2005. RADAMES: A new management approach to opera: Repertory, archives and related documents. In *Proceedings - First International Conference on Automated Production of Cross Media Content for Multi-channel Distribution*, Florence, Italy, November-December. Institute of Electrical and Electronics Engineers.

Pool, Chris and Malvina Nissim. 2016. Distant supervision for emotion detection using Facebook reactions. In *Proceedings of the Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media (PEOPLES)*, pages 30–39.

Roberts, Kirk, Michael A Roach, Joseph Johnson, Josh Guthrie, and Sanda M. Harabagiu. 2012. Empatweet: Annotating and detecting emotions on twitter. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, volume 12, pages 3806–3813, Istanbul, Turkey, May. European Language Resources Association (ELRA).

Shaver, Philip, Judith Schwartz, Donald Kirson, and Cary O'Connor. 1987. Emotion knowledge: further exploration of a prototype approach. *Journal of personality and social psychology*, 52(6):1061–1086.

Sprugnoli, Rachele. 2020. Multiemotions-it: a new dataset for opinion polarity and emotion analysis for italian. In Johanna Monti, Felice Dell'Orletta, and Fabio Tamburini, editors, *Proceedings of the Seventh Italian Conference on Computational Linguistics, CLiC-it 2020*, volume 2769 of *CEUR Workshop Proceedings*, Bologna, Italy, March 2021. CEUR-WS.org.

Strapparava, Carlo, Rada Mihalcea, and Alberto Battocchi. 2012. A parallel corpus of music and lyrics annotated with emotions. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, volume 12, pages 2343–2346, Istanbul, Turkey, May. European Language Resources Association (ELRA).

Strapparava, Carlo and Alessandro Valitutti. 2004. WordNet affect: an affective extension of WordNet. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May. European Language Resources Association (ELRA).

Ye, Lu, Rui-Feng Xu, and Jun Xu. 2012. Emotion prediction of news articles from reader's perspective based on multi-label classification. In *International Conference on Machine Learning and Cybernetics*, volume 5, pages 2019–2024, Xian, Shaanxi, China, July. IEEE.

Zhang, Min-Ling and Zhi-Hua Zhou. 2007. Ml-knn: A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7):2038–2048.

Zhao, Zhenjie and Xiaojuan Ma. 2019. Text emotion distribution learning from small sample: A meta-learning approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3948–3958, Hong Kong, China, November.

Zhou, Deyu, Xuan Zhang, Yin Zhou, Quan Zhao, and Xin Geng. 2016. Emotion distribution learning from texts. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 638–647, Austin, Texas, November. Association for Computational Linguistics.

Zoppelli, Luca. 2001. Il teatro dell'umane passioni: note sull'antropologia dell'aria secentesca. In *I luoghi dell'immaginario barocco*. Liguori, Napoli, Italia.

Appendix A: Label Distribution across Periods

Figure 1 shows the label distribution at the aria level in both AriEmozione 1.0 and AriEmozione 2.0 . We use either the year of creation or the year of the first performance to allocate each opera and produce non-overlapping bins of five years.

Tristezza is the most represented emotion in both corpora, being the most frequent in eight out of twelve periods in AriEmozione 1.0 and in all nineteen periods in AriEmozione 2.0 . *Rabbia* is the second one, being the most frequent emotion in the four periods of AriEmozione 1.0 and the second in all periods but one from AriEmozione 2.0 . *Paura* is the least represented emotion in almost all intervals of both collections. By looking at all the emotions across periods, the emotion distribution is fairly stable in both the original and extended corpus.

Appendix B: One-Versus-All Self Learning Annotation of the AriEmozione 2.0 Corpus

Here we describe in detail the process to produce the silver annotations for the instances in the AriEmozione 2.0 corpus (cf. Section 5.1). We started by merging the training and development partitions of AriEmozione 1.0 and produced six one-versus-all collections, each corresponding to one emotion with the instances belonging to the other five classes simply turned into class all. Each of the six collections is then re-partitioned into training and development partitions on an 8:2 ratio. Since we are interested in spotting the actual emotion of each new instance, we adopt precision as our single evaluation metric. The model we use is the best one from our experiments on corpus AriEmozione 1.0 (cf. Section 4.3): a 2-layer NN with TF-IDF character 3-grams.

Algorithm 1 sketches the iterative self-learning annotation process, which is applied in parallel for each of the six emotions. The input to the process includes the new training and development collections for each binary task and the raw instances, which lack annotation (lines 2–4). The output consists of the instances in the raw dataset, with emotions pre-labeled. In each iteration, baseline binary classifiers are trained on the existing labeled training data and evaluated on a fix development set (lines 7–8). The same model is applied to the set of raw instances, which are then ranked according to the classification confidence score, and the top instances are selected as candidates to join the training material (lines 9–10). Such candidates are added to the original training material at this iteration, a new model is trained from scratch, and its performance on the development set gets measured (lines 12–13). If the resulting precision is higher than

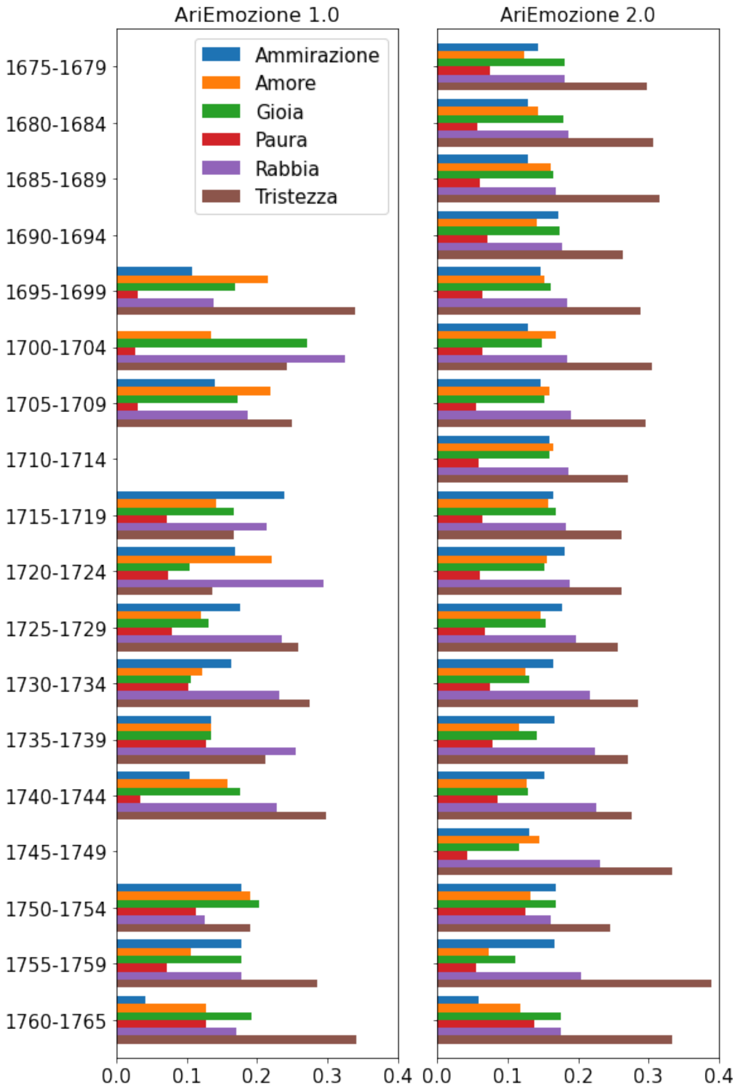


Figure 1
Emotion distribution across 5-year periods in both AriEmozione 1.0 (manual annotation; left) and AriEmozione 2.0 (automatic annotation; right) starting in 1675.

the baseline model, we transfer the new instances from the raw dataset to the training material for the next iteration (lines 15–16). Otherwise, the instances are kept in the raw set and a new iteration begins. The process runs until a minimum evaluation score is reached or the raw material gets all integrated to the training one. In our experiments, the second condition was never met. At last, 952 verses of `ammirazione`, 2,070 verses of `amore`, 1,120 verses of `gioia`, 1,320 verses of `paura`, 1,890 verses of `rabbia`, and 2,820 verses of `tristezza` were pre-selected.

Algorithm 1 Pseudo-code for the self-learning annotation process.

```
$E \leftarrow$ [amore, gioia, ammirazione, rabbia, tristezza, paura]
$tr[e] \leftarrow$ 80% of AriEmozione 1.0 , binarised as $e$ vs rest $\forall e \in E$
$te[e] \leftarrow$ 20% of AriEmozione 1.0 , binarised as $e$ vs rest $\forall e \in E$
$raw \leftarrow$ full set of fresh, non-annotated, instances
while $precision[e] < thres \forall e \in E$ or $raw \neq \emptyset$ do
    for e $\in E$ do
        $model[e] \leftarrow$ train binary classifier on $tr[e]$
        $precision[e] \leftarrow$ evaluate $model[e]$ on $te[e]$ and record the performance
        $scores[e] \leftarrow$ predict all $raw$ instances record $model[e]$'s confidence scores
        $top[e] \leftarrow$ top-$k$ instances in $raw$ with the highest confidence scores

        $model'[e] \leftarrow$ train binary classifier on $tr[e] \cup top[e]$
        $precision'[e] \leftarrow$ evaluate $model'[e]$ on $te[e])$ and record the new performance
        if $precision'[e] > precision[e]$ then
            $tr[e] \leftarrow tr[e] \cup top[e]$
            $raw \leftarrow raw \setminus top[e]$
        else
            continue
        end if
    end for
end while
```

Table 1
Accuracy and F$_1$-measure on the test set of the AriEmozione 1.0 corpus using different training partitions: AE1.0$_{tr}$=training set from AriEmozione 1.0 ; AE1.0$_{de}$=development set from AriEmozione 1.0 ; AE2.0=full AriEmozione 2.0 .

train material	Acc	F$_1$
AE1.0$_{tr}$∪AE1.0$_{de}$	0.413	0.394
AE2.0	0.417	0.411
AE1.0$_{tr}$∪AE1.0$_{de}$∪AE2.0	0.419	0.413

Appendix C: Impact of AriEmozione 2.0 on the Performance at the Verse Level

Before shifting to the aria granularity level, we performed an experiment to observe the impact of the silver data from AriEmozione 2.0 in the verse-level classification. We trained a 2-layer neural networks with TF–IDF character 3-grams (the best configuration in Table 4) on (*i*) training plus development sets from AriEmozione 1.0 , (*ii*) AriEmozione 2.0 alone, and (*iii*) the union of both. We evaluated the three models on the testing partition of AriEmozione 1.0 . We repeat each experiment three times to enhance the reliability of the results and report the arithmetic mean of the outcomes.

Table 1 shows the results. The presence of the instances from AriEmozione 2.0 , even when used alone enhance the overall performance only slightly. Still, it boosts significantly the prediction performance for some of the classes; in particular *amore* and *paura*. Table 2 shows the diagonal values of the associated confusion matrices. When the model is exposed to instances from AriEmozione 1.0 alone, the precision on both class

Table 2
Diagonal values of the confusion matrices of the predictions on the test set of AriEmozione 1.0 when the models get trained with different data partitions: AE1.0$_{tr}$=training set from AriEmozione 1.0 ; AE1.0$_{de}$=development set from AriEmozione 1.0 ; AE2.0=full AriEmozione 2.0 .

train material	ammiraz.	amore	gioia	paura	rabbia	tristezza
AE1.0$_{tr}$∪AE1.0$_{de}$	0.333	0.006	0.300	0.067	0.532	0.600
AE2.0	0.556	0.234	0.276	0.400	0.441	0.550
AE1.0$_{tr}$∪AE1.0$_{de}$∪AE2.0	0.556	0.243	0.279	0.400	0.439	0.549

amore and *paura* tend to zero. Adding the new material from AriEmozione 2.0 rises the precision on both classes 0.243 and 0.400, at the cost of a lower performance on some of the other emotions.

Leveraging Bias in Pre-trained Word Embeddings for Unsupervised Microaggression Detection

Tolúlọpẹ́ Ògúnrẹ̀mí[*]
Stanford University, United States

Valerio Basile[**]
University of Turin, Italy

Tommaso Caselli[†]
University of Groningen, Netherlands

Microaggressions are subtle manifestations of bias (Breitfeller et al. 2019). These demonstrations of bias can often be classified as a subset of abusive language. However, not much focus has been placed on the recognition of these instances. As a result, limited data is available on the topic, and only in English. Being able to detect microaggressions without the need for labeled data would be advantageous since it would allow content moderation also for languages lacking annotated data. In this study, we introduce an unsupervised method to detect microaggressions in natural language expressions. The algorithm relies on pre-trained word-embeddings, leveraging the bias encoded in the model in order to detect microaggressions in unseen textual instances. We test the method on a dataset of racial and gender-based microaggressions, reporting promising results. We further run the algorithm on out-of-domain unseen data with the purpose of bootstrapping corpora of microaggressions "in the wild", perform a pilot experiment with prompt-based learning, and discuss the benefits and drawbacks of our proposed method.[1]

1. Introduction

The growth of Social Media platforms has been accompanied by an increased visibility of expressions of socially unacceptable language online. In a 2016 Eurobarometer survey, 75% of people who follow or participate in online discussions have witnessed or experienced abuse or hate speech. With this umbrella term, different phenomena can be identified ranging from offensive language to more complex and dangerous ones, such as hate speech or doxing. Recently, there has been a growing interest by the Natural Language Processing community in the development of language resources and systems to counteract socially unacceptable language online. Most previous work has focused on few, easy to model phenomena, ignoring more subtle and complex ones, such as microaggressions (Jurgens, Hemphill, and Chandrasekharan 2019).

Microaggressions are brief, everyday exchanges that denigrate stigmatised and culturally marginalised groups (Merriam-Webster 2021). They are not always perceived as hurtful by either party, and they can often be detected as positive statements by current

[*] E-mail: `tolulope@cs.stanford.edu`
[**] E-mail: `valerio.basile@unito.it`
[†] E-mail: `t.caselli@rug.nl`
[1] Copyright ©2021 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

hate-speech detection systems (Breitfeller et al. 2019). The occasionally unintentional hurt caused by such comments is a reflection of how certain stereotypes of others are baked into society. Sue et al. (2007) define microaggressions in the racial context, particularly when directed toward people of color, as "brief and commonplace daily verbal, behavioral, or environmental indignities", such as: *"you are a credit to your race."* (intended message: it is unusual for someone of your race to be intelligent) or *"do you think you're ready for college?"* (indented message: it is unusual for people of color to succeed). The need for moderation of hateful content has previously been explored. For instance, Mathew et al. (2019b) analyses the temporal effects of allowing hate speech on Gab, a social network known for attracting a right-wing userbase, and finds that the language of users tends to become more and more similar to that of hateful users over time. Mathew et al. (2019a) further highlights that the spreading speed and reach of hateful content is much higher than the non-hateful content. As a result, being able to remove instances of hateful language, such as microaggressions, is of great importance.

Previous work on microaggressions with computational methods is quite recent. Breitfeller et al. (2019) is one of the first works to address microaggressions in a systematic way, also introducing a first dataset, SelfMA. A further contribution specifically focused on racial microaggression is Ali et al. (2020), where the authors focus on the development of machine learning systems. In terms of automatic classification, these works propose supervised methods based on linguistic features, obtaining acceptable performance but at the same time tying the results to specific benchmarks and training sets.

In this study we introduce an unsupervised method for microaggression detection. Our method utilizes the existing bias in word-embeddings to detect words with biased connotations in the message. Although unsupervised approaches tend to be less competitive than their supervised counterparts, our method is language-independent and thus it can be applied to any language for which embedding representations exist. Furthermore, the reliance of our methods on specific lexical items and their context of occurrence makes transparent the flagging of a message as an instance of a microaggression. In addition to the usefulness of our method in languages with no labeled data, the reliance of our model on words in the sentences would make it interpretable as it allows human moderators to understand what the system has based its decision on.

Our contributions can be summarised as follows:

- we introduce a **new unsupervised method** for the detection of microaggressions which builds on top of pre-trained word embeddings;
- we **further test** the proposed algorithm **on unseen data from a different domain** (i.e., Twitter), in order to qualitatively evaluate its efficacy in discovering new instances of microaggression;
- we **compare** our approach with prompt-based learning to better assess its advantages and limits.

The rest of this paper is structured as follows: we introduce our method in Section 2. The data and our results are reported in Section 3. We deploy our model and discuss its limitations in Section 4. The application of our unsupervised approach on the Twitter data and the results of this experiment are presented in Section 5. In addition to this, we further compare our method with a very recent approach, i.e., *prompt-based learning*, showing its potential advantages in Section 6. Finally, we present the conclusion and future work in Section 7.

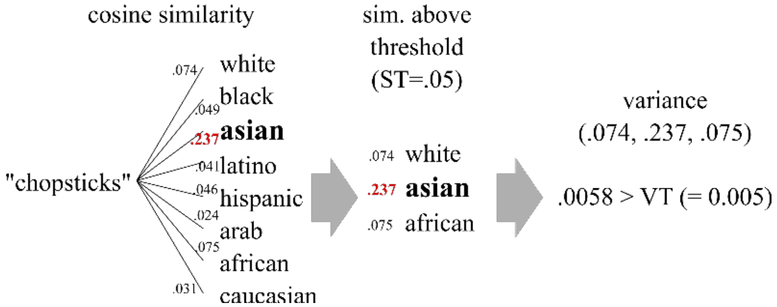


Figure 1
Worked example of unsupervised method for word "chopsticks" in the message *"Ford: Built With Tools, Not With Chopsticks"*

2. Use the Bias Against the Bias

Embedded representations, either from pre-trained word embeddings or pre-trained language models, have been shown to contain and amplify the biases present in the data used to generate them (Bolukbasi et al. 2016; Lauscher and Glavaš 2019; Bhardwaj, Majumder, and Poria 2020). As such, they often exhibit gender and racial bias (Swinger et al. 2019). Many studies have attempted to reduce this bias (Yang and Feng 2020; Zhao et al. 2018; Manzini et al. 2019). In this work, we take a different turn by using this bias to our advantage: rather than taming the hurtfulness of the representations (Schick, Udupa, and Schütze 2021), we actively use it to promote social good. In this first study, we employ word representations derived from generic textual corpora of English, in order to capture the background knowledge needed to disambiguate instances of microaggressions in the text. Recently, however, there have been studies involving word representations created from tailored collections of social media content aimed at capturing abusive phenomena like verbal aggression (Dynel 2021) and hate speech (Caselli et al. 2021).

We devise a simple and effective method that exploits existing bias in word embeddings and identify words in a message that are related to particular and distant semantic areas in the embedding space. Messages are analysed in three steps: first, for each token t^i we compute its relatedness to a list of manually curated seed words $s = s_1, ..., s_n$ denoting potential targets of microaggressions; second, we consider only the similarities of the pairs (t_i, s_j) above an empirical *similarity threshold* (ST) and compute their variance v_i; finally, we classify the token t_i as a micro aggression trigger, and consequently the message as a micro aggression, if the v_i is above an empirically determined *variance threshold* (VT).

The intuitive idea behind this algorithm is that some lexical elements in a verbal microaggression are often (yet sometimes subtly) hinting at specific features of the recipient of the message, in an otherwise neutral lexical context.

In this work, we choose to focus on microaggressions related to race and gender, therefore the seed words have to be chosen accordingly. The seed word lists for race and gender are, respectively, *[white, black, Asian, latino, hispanic, Arab, African, caucasian]* and *[girl, boy, man, woman, male, female]* for gender. There is also a practical reasons to focus on gender and race, namely the scarcity of data available for other categories of microaggression and other idiosincrasies of the available datasets — the religion class

Table 1
Statistics of the two subsets of the SelfMA dataset used in this paper, and the extra data downloaded to balance the dataset.

Source	Number of posts
SelfMA Gender	1,314
SelfMA Racial	1,278
Tumblr	2,021

was specific to different religions, therefore hard to generalise, sexuality and gender presented a large overlap, and so on.

An example of how the proposed method works is illustrated in Figure 1. In the example, consider the word "chopsticks" in the message *"Ford: Built With Tools, Not With Chopsticks"* (from the SelfMA dataset, described in Section 3). The target word exhibits a much higher relatedness to the word *Asian* (0.237) than any other seed words. Even just considering the seed words with a similarity above a fixed threshold (*white*, *Asian* and, *African*), the variance of their similarity score with respect to *chopsticks* is still higher than the variance threshold, and therefore this target word, in this context, triggers a microaggression according to the algorithm. This process is repeated for all the words in the message in order to detect microaggressions. Some categories of words are bound to exhibit a high relatedness to all the seed words, e.g., “people” or “human”. This is the reason to introduce the variance threshold in the final step of our algorithm, to filter out these cases when classifying a given message, and instead focus on words that are related to different races (or genders) unevenly, with a skewed distribution of similarity scores.

An important by-product of this algorithm is that the output is one or more trigger words, in addition to the microaggression label — in the example, the trigger word is indeed *chopsticks* — therefore enabling a more informative and interpretable decision process.

3. Experiments

To test our method, we use two subsets of the *SelfMA: microaggressions.com* dataset (Breitfeller et al. 2019), comprised of 1,314 and 1,278 Tumblr posts respectively[2]. The posts in SelfMA are all instances of microaggressions, manually tagged with one of four categories: race, gender, sexuality and religion. These posts can be tagged with more than one form of microaggressions, meaning certain instances can appear in both subsets of race and gender used for the purposes of this study. The dataset consists of first and second hand accounts of microaggressions, as well as direct quotes of phrases or sentences said to the person posting. In order to reduce linguistic perturbation introduced by accounts of a situation, we only take direct quotes found in the dataset as instances of microaggressions that we can detect with our unsupervised method. For training, we pull out direct quotes from the gender (561) and racial (519) dataset to test the algorithm. In order to balance the dataset, we scraped 2,021 random Tumblr posts, for a total of 4,612 instances. Table 1 summarises the composition of our dataset.

[2] Tumblr is a popular American microblogging platform `https://www.tumblr.com`

It is important to note that a microaggression can have multiple tags, so there is an overlap of instances. However, the seed words used to detect microaggression types in the method are different for each target phenomenon (e.g., race, gender).

We ran the algorithm on the *SelfMA* dataset, empirically optimising the two thresholds on the training split, for each word embedding type and each microaggression category, filtering by the seed words listed in Section 2. We test the algorithm with three pre-trained word embedding models for English, namely `FastText` (Joulin et al. 2017), trained on Wikipedia and Common Crawl, `word2vec` (Mikolov et al. 2013), trained on Google News, and `GloVe` (Pennington, Socher, and Manning 2014), trained on Wikipedia, GigaWord corpus, and Common Crawl. The optimization is performed by exhaustive grid search over the hyperparamter space.

To provide a better context to interpret the results, we also present the results of a simple baseline method based on the presence of seed words in the text instances. In this method, an instance is considered a microaggression if and only if any of the seed words used by the unsupervised algorithm is present in the text.

The results, shown in Table 2, indicate that `FastText` has a better F1 score on Racial microaggressions while `word2vec` performs better on Gender microaggressions. The difference in performance between `FastText` and `word2vec` is not major, and we attribute this to the difference between the corpora on which the two models were trained (i.e., web crawl and Wikipedia for `FastText` vs. news data for `word2vec`). The `GloVe` pretrained model, trained on a combination of newswire texts, encyclopedic entries and texts from the Web, underperforms in both experiments. In general, the absolute figures are encouraging, especially considering the simplicity of this unsupervised approach.

4. Limitations of Unsupervised Method

Despite promising results with the unsupervised methods, it is important to note that this method currently works on the basis of one trigger word. An analysis of the set of trigger words for each instance show that the vast majority of instances marked as microaggressions are explicitly realized, i.e., they have trigger words that are similar to or substitutes for our sets of race-related or gender-related seed words e.g Chinese, Japanese, Mexican, mister, or girlfriend. The mention of a "girlfriend" or the word "Chinese" alone in a statement should not flag it as a microaggression, so the methods needs more work to more accurately detect microaggressions with detailed reasoning. However, as the examples in Table 3 highlight, it suffice the presence of a single word to a seemingly neutral or positive statement to make it a microaggression. Examples are in Table 3.

In instances where there are multiple trigger words, the set of words selected seems to paint a picture explaining why such a word triggers a microaggression. Examples are in Table 4. In the first example, we see that the person quoted felt the need to mention that the person spoken about is "*Black*, you know", because he was *smiling*. We see something similar take place when *cute* is equated to being *feminine*.

It is possible that a method that incorporates the set of these words, or even the juxtaposition of individual words with words that don't get flagged up with the current method may lead to more precise and categorisations.

Table 2
Results of the experiment on the Gender and Racial subset of SelfMA, in terms of Precition (P), Recall (R), and F1-score (F1) on the positive class (MA), on the negative class (not-MA), and their macro-average. Best scores per microagression category are in bold.

Target	Model	Class	Precision	Recall	F1-Score
Gender	baseline	not-MA	.613	.912	.734
		MA	.825	.418	.555
		macro avg.			.644
	FastText	not-MA	.609	.746	.671
		MA	.714	.570	.634
		macro avg.			.680
	GloVe	not-MA	.692	.380	.491
		MA	.603	.848	.705
		macro avg.			.598
	word2vec	not-MA	.659	.789	.718
		MA	.769	.634	.694
		macro avg.			**.706**
Race	baseline	not-MA	.576	.950	.717
		MA	.826	.253	.388
		macro avg.			.552
	FastText	not-MA	.659	.875	.654
		MA	.814	.547	.752
		macro avg.			**.702**
	GloVe	not-MA	.765	.371	.500
		MA	.611	.896	.726
		macro avg.			.613
	word2vec	not-MA	.640	.814	.747
		MA	.776	.584	.667
		macro avg.			.692

5. Discovering Microaggressions

To better understand the performance of our unsupervised model, we performed an additional experiment. Our goal is to understand the false positive results and the potential harm the model could cause. To do so, we use our unsupervised model to label unseen instances from another domain (Twitter) than the SelfMA dataset (Tumblr) in order to see how the model would perform in detecting microaggressions.

We begin by performing keyword searches on Twitter (using Twitter's official API) and collect a new dataset of 3M tweets with seven keywords potentially containing race and gender expressions. Next, we set the threshold values ST and VT in our model in order to obtain the highest Precision scores, rather than the highest F1 value. This step is performed exactly like the optimization described in Section 2 with the only difference of the target metric. The aim of this step is to only label tweets as microaggressions with the highest possible degree of confidence. We set $ST = 0.12$ and $VT = 0.014$ for racial microaggressions leading to Precision of .931 and $ST = 0.13$ and $VT = 0.019$ for gender-based microaggressions leading to a Precision of .912. Precision has been measured on the original SelfMA dataset used as a validation set.

Table 3
Instances of microaggressions identified by one word.

Instance	Trigger word
"I've seen you around and always wanted to talk to you. You just have this wonderful... ethnicity about you."	ethnicity
"Stop acting like a princess! You're acting like a princess!! Ooh... little princess... boo hoo."	princess
"They hit a state trooper head on. And they were both illegals. Well, I don't know if they were illegals, but they had illegal sounding names."	illegals

Table 4
Instances of microaggressions identified by several words.

Instance	Trigger words
"Oh he's very nice. He's so intelligent and always happy and smiling, and very professional. (pause) He's black, you know."	smiling, black
"You like little cute dogs. That's feminine."	cute, feminine

We then run the unsupervised model on the new Twitter dataset by automatically labelling 256,843 tweets for gender and 373,631 tweets for race. After the data is labeled, we manually explore the positive instances in order to evaluate the performance of the model. The algorithm tuned for high precision found in this dataset 6,306 gender-related microaggression candidates, 13,004 race-related microaggression candidates.

We find that while the model does detect actual instances of microaggression, there is a noticeable amount of false positive instances. These tweets discuss race or gender in some manner. However, they do not necessarily contain microaggressions towards these groups. While the model does learn to detect discussions of these topics, it seems to sometimes confuse these discussions with microaggressions towards the aforementioned groups. Some examples follow, paraphrased to avoid tracking the original messages.

1. *Saying "Arrested Development isn't funny" in an office full of women just to feel something*

2. *"Men have moustaches, women have oversized bracelets"*

The humorous attempts in this tweets hinge on gender stereotypes, and therefore in some contexts it could be perceived as offensive by some recipients. The high relatedness in the word embedding space between some words (moustaches and bracelets) and gender-related seed words (men and women) triggers the detection algorithm.

The automatic detection of racial microaggressions "in the wild" is more challenging than gender-based ones, according to our manual exploration of this automatically labeled dataset. This may be due to the difficulty of crafting a list of seed words that is sufficiently race-related, but at the same time avoids generating too many false positives. We indeed found many of them, mainly due to named entities and multi-word expressions such as "White House", or simply because of the polysemy of color words, e.g. "black" and "white". We, however, still found instances of messages containing different extent of racial stereotyping, as indicated in the following examples:

3. *"why are you being so dramatic? just say I'm not originally arab, you don't have to fight about it"*

4. *"I will need to explain that to the chinese old lady who works at my school's administrative office"*

In summary, running the unsupervised microaggression detection algorithm on unseen data seems to represent a promising intermediate step towards the semi-automatic creation of language resources for this phenomenon. While the accuracy is not ideal, and lists of seed words have to be handcrafted carefully in order to avoid false positives, these drawbacks are balanced by the fairly cheap computational cost and the ease of application in a multilingual scenario.

6. Prompt-based Classification of Microaggressions

One of the advantages of the method we propose in this paper is that, being unsupervised, it allows us to perform microaggression classification in a zero-shot fashion. *Prompt-based learning* (Liu et al. 2023) is a recent paradigm which gained enormous traction in the NLP community, applied, among other tasks, to zero-shot classification. In a nutshell, prompt-based classification makes use of large pre-trained language models to map labels to handcrafted or automatically derived natural language expressions. The plausibility of the instance to classify augmented with the prompt according to the model determines the label, without the need for further training or fine-tuning.

As a final experiment on the microaggression benchmark we presented in this paper, we compute the performance of a basic prompt-based method for classification. We test two variants of prompts, one "objective" and one "subjective". The objective prompts have the form of the short sentence *"This is [mask]"* following the text of the instance to classify. *[mask]* is replaced by *offensive* and *ok*, linked respectively to the labels *MA* and *not-MA*. The subjective prompts work similarly, but the alternative template is *"I feel [mask]"* and, in order to keep the syntax consistent, the fillers for the mask are *offended* and *ok*. Table 5 summarizes the design of the prompts for this experiment.

Table 5
Objective and subjective prompts used for zero-shot microaggression classification.

Prompt type	Label	Prompt text
Objective	MA	This is offensive.
Objective	non-MA	This is ok.
Subjective	MA	I feel offended.
Subjective	non-MA	I feel ok.

The experiment is implemented with the OpenPrompt library for Python (Ding et al. 2022). The pre-trained model prompted in this experiment is the `bert-base-uncased` model based on BERT (Devlin et al. 2019). And the results

Table 6
Results of the experiment of prompt-based classification on the Gender and Racial subset of SelfMA, in terms of Precition (P), Recall (R), and F1-score (F1) on the positive class (MA), on the negative class (not-MA), and their macro-average.

Target	Prompt type	Class	Precision	Recall	F1-Score
Gender	Objective	not-MA	.823	.627	.712
		MA	.556	.776	.648
		macro avg.			.680
	Subjective	not-MA	.839	.666	.743
		MA	.587	.788	.673
		macro avg.			.708
Race	Objective	not-MA	.819	.624	.708
		MA	.540	.762	.632
		macro avg.			.670
	Subjective	not-MA	.817	.642	.719
		MA	.549	.753	.635
		macro avg.			.677

are shown in Table 6. The first observation we can draw from the results is that the subjective prompts are consistently better at predicting the correct microaggression label. While we did not systematically test a large variety of variations of prompts, this result matches the intuition that microaggression detection is a subjective task, whose perception is dependant on the recipient's perspective.

Comparing the results of the prompt-based classification with the results of the main experiment (Table 2), we see a generally comparable performance. On the gender subset, the prompt-based classification is actually slightly better in terms of macro-averaged F1-score, although the performance on the positive class (arguably more useful in a detection task) is lower. On the race subset, the classification performance is lower, although not by a large margin. Considering that we only tested fixed, hand-crafted prompts without further tuning and optimization, the results of this experiment indicate a promising application of prompt-based learning to the task of microaggression detection. On the other hand, the main unsupervised method presented in this paper retains characteristics of transparency and interpretability that are difficult to replicate with the prompt-based approach.

7. Conclusion and Future Work

In this paper we introduce a novel algorithm that exploits the existing bias in pre-trained word embeddings to detect subtly abusive language phenomena such as microagressions. While supervised methods of detection in the field of natural language processing are plentiful, these methods are only viable for languages and topics with available labeled datasets. That is however not the case for many languages. As a result, the unsupervised method of detection introduced in this study could help address the need for the moderation of microaggressions in languages other than English. This is further helped by the availability of multilingual word-embeddings as they would allow the method to be used in any of the languages supported by the embedding.

The method is unsupervised and only needs a small list of seed words. Considering its simplicity, the results obtained from an experiment on a dataset of manually annotated microaggressions are very promising. The experimental results are also compared to a recent approach based on prompt-based learning, which obtains comparable but lower performance. Further, the method is transparent, explicitly identifying the words triggering a microaggression, and thus paving the way for explainable microaggression detection.

Although the preliminary results are promising, an experiment on unseen data from a different domain shows that there is leeway for improvement. Given that we are looking at the explicit words used in each message, our method is not sensitive to implicit expressions like “you people" or “your kind", often occurring in microaggressions. We would have to add further steps to our algorithm to catch expressions like these.

Polysemy is another known issue, e.g., in words like “black" and “white" whose relatedness to certain identified trigger words could not necessarily be due to race. While a careful composition of the seed word lists helps to minimize this issue, a systematic approach to polysemy would certainly be desirable. The seed word list may also be expanded, either manually or exploiting existing lexicons such as HurtLex (Bassignana, Basile, and Patti 2018) for offensive terms (including stereotypes for several categories of individuals) or specialized lists of identity-related terms[3].

In future work, we plan on improving our model to account for lexical ambiguity, and the complexity derived from the interference between pragmatic phenomena and aggression, e.g., in humorous and ironic messages, following the intuition in recent literature (Frenda 2018) about the interconnection between irony or sarcasm and abusive language online. Our current plan is to apply the algorithm presented in this paper to bootstrap the creation of a multilingual resource of online verbal microaggressions and release it to the research community.

Acknowledgements

This work of Valerio Basile is partially funded by Compagnia di San Paolo - Bando expost 2020 - “Toxic Language Understanding in Online Communication - BREAKhateDOWN”.

References

Ali, Omar, Nancy Scheidt, Alexander Gegov, Ella Haig, Mo Adda, and Benjamin Aziz. 2020. Automated detection of racial microaggressions using machine learning. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 2477–2484, Canberra, Australia, December 1–4. IEEE.

Bassignana, Elisa, Valerio Basile, and Viviana Patti. 2018. Hurtlex: A multilingual lexicon of words to hurt. In *5th Italian Conference on Computational Linguistics, CLiC-it 2018*, volume 2253, pages 1–6, Torino, Italy, December 10–12. CEUR-WS.

Bhardwaj, Rishabh, Navonil Majumder, and Soujanya Poria. 2020. Investigating gender bias in bert. *Cognitive Computation*, 13:1008–1018.

Bolukbasi, Tolga, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS’16, pages 4356–4364, Barcelona, Spain, December 5–10. Curran Associates Inc.

[3] See for instance this compendium of LGBTQIA+ terminology: `https://www.umass.edu/stonewall/sites/default/files/documents/allyship_term_handout.pdf`

Breitfeller, Luke, Emily Ahn, David Jurgens, and Yulia Tsvetkov. 2019. Finding microaggressions in the wild: A case for locating elusive phenomena in social media posts. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1664–1674, Hong Kong, China, November 3–7. Association for Computational Linguistics.

Caselli, Tommaso, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2021. HateBERT: Retraining BERT for abusive language detection in English. In *Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021)*, pages 17–25, Online, August. Association for Computational Linguistics.

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

Ding, Ning, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Haitao Zheng, and Maosong Sun. 2022. OpenPrompt: An open-source framework for prompt-learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 105–113, Dublin, Ireland, May. Association for Computational Linguistics.

Dynel, Marta. 2021. Humour and (mock) aggression: Distinguishing cyberbullying from roasting. *Language & Communication*, 81:17–36, November.

Frenda, Simona. 2018. The role of sarcasm in hate speech. a multilingual perspective. In E. Lloret, E. Saquete, P. Martinez-Barco, and I. Moreno, editors, *Doctoral Symposium of the XXXIVInternational Conference of the Spanish Society for Natural Language Processing (SEPLN 2018)*, pages 13–17, Seville, Spain, September 19–21.

Han, Xu, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Yuan Yao, Ao Zhang, Liang Zhang, Wentao Han, Minlie Huang, Qin Jin, Yanyan Lan, Yang Liu, Zhiyuan Liu, Zhiwu Lu, Xipeng Qiu, Ruihua Song, Jie Tang, Ji-Rong Wen, Jinhui Yuan, Wayne Xin Zhao, and Jun Zhu. 2021. Pre-trained models: Past, present and future. *AI Open*, 2:225–250, January.

Joulin, Armand, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain, April 3–7. Association for Computational Linguistics.

Jurgens, David, Libby Hemphill, and Eshwar Chandrasekharan. 2019. A just and comprehensive strategy for using NLP to address online abuse. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3658–3666, Florence, Italy, July 28–August 1. Association for Computational Linguistics.

Lauscher, Anne and Goran Glavaš. 2019. Are we consistently biased? multidimensional analysis of biases in distributional word vectors. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 85–91, Minneapolis, Minnesota, June 6–7. Association for Computational Linguistics.

Liu, Pengfei, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys (CSUR)*, 55:1–35, September.

Manzini, Thomas, Lim Yao Chong, Alan W. Black, and Yulia Tsvetkov. 2019. Black is to criminal as caucasian is to police: Detecting and removing multiclass bias in word embeddings. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 615–621, Minneapolis, Minnesota, June 2–7. Association for Computational Linguistics.

Mathew, Binny, Ritam Dutt, Pawan Goyal, and Animesh Mukherjee. 2019a. Spread of hate speech in online social media. In *Proceedings of the 10th ACM Conference on Web Science*, WebSci '19, page 173–182, New York, NY, USA, May 27–30. Association for Computing Machinery.

Mathew, Binny, Anurag Illendula, Punyajoy Saha, Soumya Sarkar, Pawan Goyal, and Animesh Mukherjee. 2019b. Temporal effects of unmoderated hate speech in gab. *CoRR*, abs/1909.10966, September.

Merriam-Webster. 2021. Merriam-webster's definition of microaggression. `https://www.merriam-webster.com/dictionary/microaggression`. Accessed: 2021-03-08.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., December.

Pennington, Jeffrey, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 25–29. Association for Computational Linguistics.

Schick, Timo, Sahana Udupa, and Hinrich Schütze. 2021. Self-diagnosis and self-debiasing: A proposal for reducing corpus-based bias in NLP. *Transactions of the Association for Computational Linguistics*, 9:1408–1424, December.

Sue, Derald, Christina Capodilupo, Gina Torino, Jennifer Bucceri, Aisha Holder, Kevin Nadal, and Marta Esquilin. 2007. Racial microaggressions in everyday life: Implications for clinical practice. *The American psychologist*, 62:271–86, May.

Swinger, Nathaniel, Maria De-Arteaga, Neil Thomas Heffernan IV, Mark D.M. Leiserson, and Adam Tauman Kalai. 2019. What are the biases in my word embedding? In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 305–311, Honolulu, HI, USA, January 26–27.

Yang, Zekun and Juan Feng. 2020. A causal inference method for reducing gender bias in word embedding relations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9434–9441, New York City, USA, February 7–12.

Zhao, Jieyu, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang. 2018. Learning gender-neutral word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4847–4853, Brussels, Belgium, October 31–November 4. Association for Computational Linguistics.

Word Usage Change and the Pandemic: A Computational Analysis of Short-Term Usage Change in the Italian Reddit Community

Edoardo Signoroni[*]
Università di Pavia, Masaryk University

Elisabetta Jezek[**]
Università di Pavia

Rachele Sprugnoli[†]
Università di Parma

The COVID-19 pandemic has affected every aspect of our lives. Our work assesses whether it has also impacted the usage of the Italian language, particularly its lexicon. We create a new corpus of Italian texts taken from Reddit and apply a recent unsupervised usage change detection method on two sub-corpora, one with data from 2019 and one with data from 2020. The focus of our investigation is short-term usage change. The results for the first 10-top candidates and for a selection of candidates among the top-100 are analyzed, to show that usage change has indeed happened.

1 Introduction

Each aspect of language changes over time, but meaning is the one more susceptible to mutation. According to Blank (1999), change is only a side-effect of the speakers' pragmatic goal, which is to achieve success in communication. This also means that change is a consequence of the human mind and social interactions: innovations are thus employed and adopted because they are judged to be the most successful strategy to communicate effectively.

The study of meaning change was the focus of the first scholars of semantics but while they employed manual methods, nowadays many studies are conducted with automatic and semi-automatic tools stemming from computational linguistics and computer science.

Lexical Semantic Change (LSC) detection, which aims at identifying the change in meaning of words over time using corpus data, is a Natural Language Processing (NLP) task pertaining to lexical and diachronic semantics. Recently, this field has seen an exponentially rising interest but work for languages other than English is still relatively scarce (Schlechtweg et al. 2020).

The computational literature approaches the task in several ways and with different terminologies: Tahmasebi, Borin, and Jatowt (2021) define the field as "lexical semantic change detection"; this definition is also adopted by both Schlechtweg et al. (2020) and

[*] NLP Centre, Faculty of Informatics - Botanická 68a, 602 00 Brno, Czech Republic. E-mail: e.signoroni@mail.muni.cz

[**] Dipartimento di Studi Umanistici, Corso Strada Nuova 65, 27100 Pavia, Italy. E-mail: jezek@unipv.it

[†] Dipartimento di Discipline Umanistiche, Sociali e delle Imprese Culturali, Via M. D'Azeglio, 85, 43125 Parma, Italy. E-mail: rachele.sprugnoli@unipr.it

Basile et al. (2020b), which set the task as "identifying words that change meaning over time". Kutuzov et al. (2018), instead, formalize the task as "detecting semantic shifts". Finally, Del Tredici, Fernández, and Boleda (2019) employ "short-term meaning shift", while Gonen et al. (2020) frame the task as "detecting usage change": in this paper we follow this latter definition.

Most of the work in this field studies meaning change across decades or even centuries, by leveraging data from different corpora of literary or newspaper data. Fewer studies investigate short-term usage change, by comparing texts produced in smaller time spans, from one to less than ten years apart. This kind of research often uses data from social media, like Twitter or Reddit. When considering such smaller time frames, it is more sensible to talk about "usage change" rather than "meaning change" of a word, as proposed by Gonen et al. (2020).

The focus on use is motivated by the distributional method adopted to investigate the data (Harris 1954), which derives information about the meaning of a word from its context of use, and assumes that words with similar distributional properties have similar meanings (Sahlgren 2008; Ježek 2016; Lenci 2018; Jurafsky and Martin 2021). The kind of semantics that stems from the distributional hypothesis is called distributional semantics or, more specifically, vector space semantics, because it represents words and their meaning as vectors in a geometric space, and calculates the similarity between vectors using the cosine function. With cosine similarity, the nearest neighbors, i.e. the items with the highest similarity score with respect to the target word, can be identified (Lenci 2018). In distributional semantics, there are several types of vectors that are computed with different methods, in particular count-based vectors obtained by counting the co-occurrences of words, and embedded vectors (called *embeddings*) obtained with predictive neural models. The vectors we use in our experiment belong to the second type and are computed with the *Skip-Gram with Negative Sampling* (SGNS) version of the *word2vec* neural model (Mikolov et al. 2013).

In our study we consider a short-term time span that represents a peculiar socio-cultural and chronological context, the pandemic. Our work starts from the hypothesis that an event such as the COVID-19 pandemic would bring forth changes in the use of words. We focus on Italian, since many other studies were done for the English language. To achieve our goal, we create a new corpus of texts from Reddit, and partition the corpus in two datasets, one for the year 2019 and one for the year 2020. After cleaning and lemmatizing the corpus, we apply the method outlined in Gonen et al. (2020) (§3.3) to our data to detect word candidates that may have undergone usage change from one dataset to the other.

Our analysis of the proposed candidates indicates that some degree of usage change has occurred: specific word senses gained prominence and new words arose as the need to express concepts connected to the pandemic became more widespread.

The paper is divided into five sections: Section 2 reviews the contribution of computational linguistics and Natural Language Processing (NLP) to the COVID-19 pandemic; it also formally defines the task of unsupervised meaning change detection and surveys different approaches. Section 3 details the methodology of this study and describes the features of our datasets. Section 4 presents the results of the work and analyzes them. Section 5 draws some conclusions.

Contributions

Our work contributes to the research on usage change detection and on the impact of the COVID-19 pandemic on language as follows:

- creating a new corpus of social media texts for Italian, focusing on short-term usage change. The corpus is available online;[1]
- testing the application of a relatively recent and computationally light method of usage change detection, previously untested for Italian;
- analyzing the impact of the pandemic on word use in a language different than English.

2 Related work

In this section we first provide an overview of the work done by the Computational Linguistics and NLP community in response to the COVID-19 pandemic (§2.1). Then, we briefly survey previous studies and methods of computational detection of meaning change (§2.2), with a focus on short-term change (§2.2.1) and on Italian (§2.2.2).

2.1 Computational Linguistics and the COVID-19 Pandemic

The Computational Linguistics and NLP community can support the research to fight the Coronavirus and its consequences by tapping into the great quantities of unstructured text and speech data; analyzing the countless published research papers, social media post and news articles can be critical to support best practices in clinical management; to understand the public response to the outbreak; to find and contrast spreading misinformation; to automatically identify and organize helpful information from the web.

One of the first resources on the COVID-19 pandemic is CORD-19, a COVID-19 Open Research Dataset[2] curated at the Allen Institute for AI in March 2020. In the same month, the 'Lab Task 1' at CLEF (Conference and Labs of the Evaluation Forum) 2020[3] asked to rank a stream of tweets on different topics, including COVID-19, according to their check-worthiness. A check-worthy tweet includes a claim that is of interest to a large audience or that might have a harmful effect. Again, in March, the Kaggle platform[4] started to organize tasks to develop text and data mining tools that can help the medical community to develop answers to high priority scientific questions. These are based on the aforementioned CORD-19 corpus, as is the TREC (Text Retrieval Conference)-COVID program[5], a challenge that follows the TREC assessment process to evaluate search systems.

In July 2020, the 1st Workshop on Natural Language Processing for COVID-19 was held at the Association for Computational Linguistics (ACL) conference. The second part of the workshop was held in the same year at the Empirical Methods in Natural Language Processing Conference (EMNLP). Both workshops demonstrated the help that the NLP community can provide, mainly in navigating the literature on the virus, in identifying and fighting misinformation and in characterizing the public reaction through the analysis of data from social media, like Twitter and Reddit.

One of the first contributions of the Italian NLP community to fight the pandemic is 40twita, part of the larger TWITA project ongoing at the University of Turin since

[1] https://github.com/edoardosignoroni/usage_change_ITA
[2] https://www.semanticscholar.org/cord19/download
[3] https://clef2020.clef-initiative.eu/
[4] https://www.kaggle.com/allen-institute-for-ai/CORD-19-research-challenge
[5] https://ir.nist.gov/covidSubmit/

2012. TWITA is a collection of tweets in Italian, first published in 2013 with about 100 million tweets from February 2012 to February 2013; the automatic collection is still ongoing. 40twita is a subset of TWITA; the dataset is collected daily from 1 March 2020 by filtering TWITA with COVID-19 related keywords.[6]

In April 2020, the "Covid-19 Semantic Browser" was developed by the Area Science Park in collaboration with the Italian Association of Computational Linguistics (AILC): it employs state-of-the-art neural networks to search relevant articles in the CORD-19 dataset.[7] Another useful tool developed by the Italian community is the "COVID19 Infodemics Observatory"[8] at the Complex Multilayer Networks (CoMuNe) Lab of the Fondazione Bruno Kessler in collaboration with Harvard's Berkman Center for Internet & Society and with IULM University in Milan. According to the WHO,[9] the pandemic has been accompanied by a massive surge of information, dubbed "infodemic", which can potentially contain inaccurate, fake, or harmful information. This makes it hard for people to find reliable and trustworthy news and sources. FBK's Observatory monitors millions of tweets with machine learning techniques to quantify collective sentiment and psychology, presence of social bots[10] and news reliability to find that almost 30% of the news are unreliable.[11]

2.2 Automatic Language Change Detection

Formally, the task of detecting meaning change can be formulated as follows: given corpora $[C_1, C_2, ..., C_n]$ containing texts created in time periods [1, 2, ..., n], the task is to locate the same words with different meaning in different time periods, or to locate the words which changed the most. Related tasks are to discover general trends in meaning change or the dynamics of the relationships between words (Kutuzov et al. 2018).

At the word level, most of change detection methods employ vectors, both count-based and neural ones (embeddings), for the words. This comes to the cost of representing all senses of a term with a single representation. Most of the count-based approaches start by building a co-occurrence matrix, often reducing its dimensions by SVD (Singular Value Decomposition). PMI (Pointwise Mutual Information) scores are used for co-occurrence strength rather than raw frequency, while vector similarity is measured with the cosine (Tahmasebi, Borin, and Jatowt 2021). Low similarity is understood as higher amount of change or polysemy.

Sagi, Kaufmann, and Clark (2009) employ context vectors, that is, the combined vectors of the words in a context window around the word under examination, while Gulordava and Baroni (2011), and Rodda, Senaldi, and Lenci (2017) use also PMI. Kahmann, Niekler, and Heyer (2017) compare changes in context similarity between ranked series at different points in time. Tang, Qu, and Chen (2013) and Tang, Qu, and Chen (2016) use contextual entropy and reduce dimensions on the fly rather than through SVD. Most of these methods are evaluated qualitatively on a random or manually selected sample.

[6] http://twita.di.unito.it/dataset/40wita

[7] http://covidbrowser.areasciencepark.it/

[8] https://covid19obs.fbk.eu/#/

[9] https://www.who.int/emergencies/diseases/novel-coronavirus-2019/situation-reports, Situation Report 13, 2 Feb 2020.

[10] A social bot is an automated computer program that interacts with users on social media.

[11] https://covid19obs.fbk.eu/#/

The works that use word embeddings train them independently over different time-sliced corpora and then compare them by projecting all representations onto the same space. More specifically, there are three main methods: i. vectors for the first time period are trained without any other information, then the representation for the successive time spans is initialized with the values of the previous interval to which they are then compared using cosine similarity to detect the change (Kim et al. 2014); ii. words are projected using linear mapping on the last time period (Kulkarni et al. 2014; Hamilton, Leskovec, and Jurafsky 2016); iii. mapping is avoided all together by comparing second order similarity, and meaning is modelled as the linear combination of the neighbors of a word from previous time points (Eger and Mehler 2016).

Dynamic word embeddings are another embedding method for meaning change detection. While different techniques exist involving these vectors, all of them train this kind of word embeddings in the same original space, and then share data across all time periods to update the word representations. Dynamic word embeddings have been shown to be beneficial, because they reduce the need of aligning independently trained embeddings, and the necessity of large datasets, rarely available for historical corpora (Tahmasebi, Borin, and Jatowt 2021). This approach is employed by Bamler and Mandt (2017), Yao et al. (2018), and Rudolph and Blei (2018). Context vectors are shared across all the time slices, while the embeddings are trained only within a single time span. It was shown that dynamic word embeddings perform better than the baselines (Tahmasebi, Borin, and Jatowt 2021).

2.2.1 Short-term Change

As mentioned in Section 1, some studies focus on investigating short-term meaning change, mostly employing textual data from social networks. Stewart et al. (2017) present a study on short-term change during the Russia-Ukraine crisis of 2014-2015, through data from VKontakte, a popular social media in the area. The aim of the research is to visualize and predict change in a word's semantics over the weeks, leveraging distributional representations. First, the tf-idf score for each word is extracted and concatenated in a time series that represents a concept drift, a measure which the authors define as a combination of a word's change in meaning and frequency. Then, temporal word embeddings are learned with the gensim implementation of word2vec: the vectors are initialized with the vocabulary of all the words in the data above a fixed frequency threshold, and then trained with tokenized posts for each weekly timestamp to generate a time series. Different word vectors are compared using cosine similarity and by looking at their neighbors. According to the authors, this study provides a generalizable proof of concept for future studies on short-term shift in social media.

Del Tredici, Fernández, and Boleda (2019) present an exploration of meaning shift within a period of 8 years with data from online community of speakers (sports subreddits), which allows better observation of short-term meaning shift. Previous research by Del Tredici and Fernández (2018) showed that this and similar communities have features that favor linguistic innovation. The behavior of a standard distributional model is tested when applied to short-term shift, showing that the model is confused by contextual changes due to particular references to people and event. A large sample of community-independent language is used to initialize the word vectors; then, these representations are updated for a certain point in time with the subreddit data.

Gonen et al. (2020) propose an alternative method to detect usage change. Specifically, they propose to work in the shared vocabulary space with the underlying intuition that words whose usage has changed are likely to be interchangeable with different sets of words. Thus, these words will have different neighbors in the embeddings spaces of the two time periods. Their algorithm first represents each word in a corpus as the set of its top k nearest neighbors; then, it computes the score for word usage change across corpora by considering the size of the intersection of the two sets of neighbors. This method will be further discussed in Section 3.3., as it is the method that we selected for our experiment.

Guo, Xypolopoulos, and Vazirgiannis (2022) apply the method proposed by Hamilton, Leskovec, and Jurafsky (2016) to a corpus of tweets posted between April and June 2020. They compute *word2vec* word embeddings for each month, using the pre-trained twitter-200 gensim[12] model as reference. Then, they align the three obtained vector spaces to track usage change, and they present four case studies: *racism*, *hero*, *quarantine*, *ai*. Even in this small sample, the shift towards words related with COVID and healthcare is tangible: *racism* shifts away from *sexism* and *homophobia* towards *asians* and *sinophobia*. *hero* moves from *veteran* and *superman* towards *frontliner* and *covidwarrior*. *quarantine* goes from *swineflu* and *flu* to *coranatine* and *corona*. *ai* moves away from *math* and *data*, towards *ehealth* and *bloodtesting*. The authors also computed the stability distribution of words between the pre-COVID-19 reference and each of the three COVID-19 models, taking the average value as its final stability measure. They conclude that the meaning change across corpora is more significant than that over monthly time periods. To the best of our knowledge, this is the only study on English that has similar objectives to ours. However, there are some methodological differences: i. they follow the alignment approach of Hamilton, Leskovec, and Jurafsky (2016) to tracking usage change; ii. they focus on an arbitrarily selected group of key-words. For this reason the results are not fully comparable to ours.

2.2.2 Works on Italian

While the majority of the experiments on meaning change detection focuses on English, there are also studies for other languages. For example, SemEval 2020 Task 1 (Schlechtweg et al. 2020) addresses the unsupervised detection of meaning change in text corpora of German, English, Latin and Swedish.

As for Italian, some research has been conducted and presented at EVALITA 2020, under the DIACR-Ita: Diachronic Lexical Semantics task (Basile et al. 2020b). The corpus from Basile et al. (2020a), divided into two sub-corpora for the years 1945-1970 and 1990-2014, was used for the DIACR-Ita task. Several methods were submitted: Post-alignment, Joint Alignment, Contextual Embeddings, Graph-based and PoS tag features. Post-alignment systems first train static embeddings and then align them, while Joint alignment does these two processes at the same time. Contextual embeddings systems are based on contextualized embeddings, such as BERT (Devlin et al. 2019). Graph-based systems rely on graph algorithms, while PoS tag features systems use the distribution of targets PoS tags across the time slices. The majority of these systems use cosine distance as a measure of meaning change, except for Contextual embedding representations and Graph-based methods (Basile et al. 2020b). The best methods (Prazák, Pribán, and Taylor 2020; Kaiser, Schlechtweg, and im Walde 2020) use Skip-Gram with Negative Sampling (SGNS) to compute word embeddings, which are then aligned. Cosine similarity and a threshold are used to detect changed words.

Basile et al. (2016) employ Temporal Random Indexing, an embedding method first used in Basile, Caputo, and Semeraro (2014). The dataset is the Italian portion of

[12] https://radimrehurek.com/gensim/

the Google Books Ngrams corpus, split into 10-year period sub-corpora for the time between 1850 and 2012. The vocabulary of each split vocabulary is modeled as the sum of its random vectors and then normalized to give less weight to the most frequent words. To detect shifts, the method by Kulkarni et al. (2014) is used. The study also employs temporal indexing to detect the average span of change in years (Tahmasebi, Borin, and Jatowt 2021).

Cafagna, De Mattei, and Nissim (2020) study how words are used differently in two Italian newspapers with diverging political opinions, *La Repubblica* (left-leaning) and *Il Giornale* (right-leaning). They focus on synchronic change, but the methodology is still relevant to the study of short-term usage change. The embeddings are first trained on *La Repubblica* texts and then updated with those from *Il Giornale*. The measure of the shift that the same word has undergone is then computed. A value for the frequency and a combination of both frequency and shift measure is also calculated. Starting from a shared vocabulary, the study features a top-down analysis, concerned with the change affecting the most frequent words in both newspapers; and a bottom-up analysis, that observes how a single word's usage varies across the two spaces looking both at its embeddings and frequency. It is proposed that the most interesting cases are those whose relative frequency does not change much in the two datasets, but still exhibit a high degree of change.

3 Methodology

This Section illustrates the methodology of our study: the creation of the corpus (§3.1) and its preprocessing (§3.2), as well as the details of the usage change algorithm we employed (§3.3).

3.1 Corpus

The dataset for this study is a newly created corpus of texts taken from Reddit.[13], a large on-line community made by more than 2.5 million user-created sub-communities called subreddits or subs.[14] As of December 2020, Reddit was the 18th-most visited website in the world, but it is still a mainly American phenomenon, with 41% of its traffic coming from the US,[15] where it is the 5th-most visited site.[16]. However, an active Italian community is present and is aggregated in a subreddit called *r/italy* from which we downloaded the texts composing our dataset.[17] Reddit gives free and easy access to historical data thus we were able to download posts (also known as submissions) and comments in the same specific time frame for the years 2019 and 2020, that is between January 30 and November 30. January 30 was chosen as the starting date of our period of interest because in 2020 it was the day when the first cases of COVID-19 were recorded in Italy. On the basis of these two time frames, the corpus is divided in two sub-corpora, one for each year (2019 and 2020).

We automatically built the corpus using a new Python 3 scraper script that allows accessing subreddit data through the Reddit API (Application Programming Interface). The Python implementation used in the scraper script is called PRAW (Python Reddit

[13] https://www.reddit.com/
[14] https://frontpagemetrics.com/history (as of December 2020).
[15] https://www.alexa.com/siteinfo/reddit.com
[16] https://www.redditinc.com/press (Retrieved December 30, 2020)
[17] As of April 2021, *r/italy* had 300,000 subscribers.

API Wrapper).[18] However, using PRAW it is not possible to download posts or comments older than the last 1000 due to limitations in the Reddit API. To overcome this limitation, another API wrapper, called PSAW (Python Pushshift.io API wrapper), was used on top of the standard one.[19] This API leverages the pushshift.io[20] database for comment and submission search. Pushshift.io is a big-data storage and analytics project which copies data and metadata when they are posted on Reddit. The project also hosts monthly dumps of comments and submissions. These features make this project very useful for analyzing large quantities of Reddit data and, crucially, allows for the retrieval of data for a specific time range.

Our script was run two times, one for each time span. The script proceeds in the following manner: it first retrieves from Pushshift.io the IDs of all submissions in *r/italy* from the newest to the oldest; it then uses PRAW to collect the title of the submission, its text, and comments. A typical submission includes a title and a more articulated text; the discussion in the comment section is nested, as every user can answer to each comment. During the scraping, the raw text is iteratively saved in a text file for each day of the time frame; the texts are organized in two symmetrical folders. To ensure anonymity, no metadata regarding the author of the submission or comment is requested or saved in any way.

Despite being a very useful resource, that is, the basis for one of the few studies on Italian and the pandemic, and the first focusing on short-term usage change, the corpus we created has some limitations:

- Multiple languages: the majority of the downloaded texts are written in Italian, however there are some posts and comments in English. These tend to occur in the same context and submissions: most of them are posts from non-Italian users which are answered and discussed in English.
- Representativeness: as a 2016 American study showed, it should be noted that, as a whole, the userbase of Reddit is not representative of the overall population. Users of Reddit were once described as "offbeat, quirky, and anti-establishment".[21] This skewed demographic characterises also the number of users of *r/italy*: the userbase of the subreddit at the moment of the creation of the corpus was of 267,306 users, which is the 0.45% of the Italian population.[22]
- Accuracy of the texts: the Pushshift.io project API copies the submission at the moment of its creation on Reddit and does not update it. However, users often modify their posts and comments. These so called "EDITs" can be quite long and elaborate at times, and thus their absence may mean some loss of useful data. Moreover, some duplicate texts are present even if some specific restrictions were included based on the structure of Reddit

[18] https://github.com/praw-dev/praw

[19] https://github.com/dmarx/psaw

[20] https://pushshift.io/

[21] https://www.journalism.org/2016/02/25/reddit-news-users-more-likely-to-be-male-young-and-digital-in-their-news-preferences/; https://www.nbcnews.com/tech/tech-news/hipster-internet-favorite-reddit-may-have-lose-its-edge-go-n824866

[22] The total population of Italy as of 1 January 2019 was 59,641,488 inhabitants (http://dati.istat.it/Index.aspx?QueryId=18460).

discussions. For instance, text of stickied posts[23] and comments are copied only once.

Despite these limitations, the resulting corpus has proved very useful to our purpose of detecting word usage change. Table 1 gives the size of the corpus and its subcorpora, both in terms of number of raw tokens and unique lemmas.

Table 1
Statistics about the corpus. The first column reports the number of white space-separated entities in the raw text files, the second column the number of unique tokens in the lemmatized text.

Year	Days	Raw Tokens	N. of Lemmas	Size
2019	305	24,141,080	283,570	151MB
2020	306	39,728,203	380,146	250MB

3.2 Pre-processing

We pre-processed the corpus following a two-step procedure: first we cleaned the texts and then we performed tokenization and lemmatization.

More specifically, the first step consisted in lowercasing all texts and removing URLs, special characters and stopwords.[24] Some special characters, however, were not removed in order to preserve specific features of Reddit, such as the use of / in the names of users and subreddit names, or the sarcasm tag /s. Words longer than 24 characters were substituted with the label "LONG" and a double paragraph break was added every 3,000 words to ease computation.

The second pre-processing step involved tokenization and lemmatization using Stanza (Qi et al. 2020). We chose Stanza because of its good performances on Italian texts: in particular, UD_Italian-ISDT is the model for which the highest accuracy is reported (97.79% for tokenization and 98.01% for lemmatization) compared to the other available models for Italian.[25] We performed lemmatization because it allows to focus on lexical meaning, removing morphological variations. The task also provided some interesting insights on problems that arise when lemmatization is applied to morphologically fusional languages, such as Italian. Indeed, a manual inspection of the processed data revealed that the lemmatizer, while performing well in the majority of the instances, had some problems with relatively uncommon words, borrowings, verbal forms, and named entities (e.g. names of states and nationalities or proper nouns and surnames). Other errors are due to non-standard spelling and form, or are the result of imprecise tokenization. Moreover, having employed an Italian lemmatization model, English words are not properly managed, for example they are often lemmatized by using Italian forms (e.g. vaccine, lemmatized as **vaccina*).

To get a rough estimation of the lemmatization quality, we compared a subsample of the lemmatized text against a list of valid Italian word-lemma pairs. After scoring 10 subsamples, we observed an average accuracy of 86%, well below the reported

[23] That is, post and comments that are fixed in place by the subreddit moderators at the top of the page.
[24] We used the Italian stop words of NLTK (https://www.nltk.org/).
[25] https://stanfordnlp.github.io/stanza/performance.html

performances. We also repeated the experiment using SpaCy[26] achieving the same accuracy. We tried to improve the quality of lemmatization in several ways, for example by adding more stopwords, removing rare words (i.e. with less than 5 occurrences), or by normalizing the spelling of words to their most frequent form. However, these attempts did not result in a significant improvement of the final results. These additional experiments confirmed that lemmatization is a complex task when applied to morphologically complex languages. The problems are even more evident when dealing with noisy non-standard texts, such as spontaneous social network conversations, for which even state-of-the-art models, which in our case are trained mostly on news corpora, struggle to cope with.

3.3 Usage Change Detection

We adopted the method from Gonen et al. (2020), introduced in §2.2.1, to detect usage change. This method is perfectly in line with the aim of our study, that is to analyze differences between corpora by detecting words that are used differently across them. The task is defined by the authors as follows: given two corpora with substantial overlapping vocabularies, identify candidate words whose predominant use is different in the two corpora. The expected result is a ranked list of words, from the one that is most likely to have changed, to the least likely.

In other words, Gonen et al. (2020) propose to work in the shared vocabulary space with the underlying intuition that words whose usage changed are likely to be interchangeable with different sets of words, and so to have different neighbors in the two embedding spaces. Their algorithm represents each word in a corpus as the set of its top k nearest neighbors. Then, it computes the score for word usage change across corpora by considering the size of the intersection of the two sets.

Words with a smaller intersection are ranked higher as candidates for usage change. It is important to note that this method only considers the words in the intersection of both vocabularies, as words that are rare in one of the corpora are easily spotted by using their frequency in the two spaces, and do not fit the definition of usage change according to the authors. This method does not require extensive filtering of words; they instead filter words based on frequency, using a large value of $k = 1000^4$, because large neighbor sets are more stable.

The advantages of this method are plenty: (i) simplicity, since there is no need for space alignment, hyperparameter tuning and vocabulary filtering; (ii) interpretability, provided by the intuitive ranking system used for providing the results; (iii) locality, with the score for each word determined only by its own neighbors (whereas in the projection methods the similarity depends on the projection itself, which implicitly takes into account all the other words and their relations); (iv) stability, because the method produces similar results across different embeddings trained on the same corpora (this is not the case for alignment-based approaches). As the authors note, however, their method still has some limitations: it assumes high quality embeddings, and so, a large corpus. This is somewhat mitigated by the fact that the minimal input required is raw text without the need of annotation. In fact these are just minimal requirements; as already mentioned in §3.2, we used lemmatized text, where each token was substituted for its lemma. Another limitation is the fact that like previous approaches, this method does not guarantee that the detected words have indeed undergone usage change but

[26] https://spacy.io/

it at least aims to highlight candidates for later human verification and interpretation (Gonen et al. 2020).

To adapt this method to our Italian corpus we firstly collated all the text of the two sub-corpora in two different text files, one for each year. The algorithm was then applied to these files, both in unlemmatized and lemmatized form, without altering any of its original parameters. The algorithm then computed the *word2vec* embeddings for both input files and returned the list of top-100 words which most likely have undergone usage change. Despite the imperfect results of lemmatization, we decided to focus our analysis on lemmatized text in order to reduce the problems connected to data sparsity and the morphological complexity of the Italian language. To visualize the output, we used t-SNE (t-distributed stochastic neighbor embeddings) as implemented in the method provided by Gonen et al. (2020) for the top-10 candidates, but we scaled down the number of represented neighbors of each word to enhance readability (see figure 1 as an example of the visualization).

4 Results and Discussion

In this Section we present and analyses the results obtained by the application of the usage change detection algorithm to the two sub-corpora.

4.1 Top-10 detected words

Table 2 lists all the top-10 neighbors in the 2019 and 2020 vector spaces for the top-10 candidate words detected by applying the Gonen et al. (2020) algorithm to our data.

The top-10 candidates can be divided in three broadly defined classes according to their nearest neighbors, and to how they have changed between the two sub-corpora. The fist class, *narrowing*, denotes candidates which changed from a more general usage, to a more specific one, but which is already present in the language. This is the case with *positivo*, *intensivo*, *guarire*, *gene*. The second class, *shift*, refers to those candidates which usage switched between two different semantic fields. Candidates such as *virus*, *testare*, *influenza* fall under this category. The last class, *not informative*, comprises those candidates which neighbors in both corpora, either due to their low frequency or noise, do not allow for a clear indication of usage. *bla*, *eco*, and *leve* are examples of not informative candidates proposed by the algorithm.

The word *positivo* (“positive”) is the first on the list. In the 2019 sub-corpus this adjective occurs mainly with terms pertaining to subjective evaluation (e.g. *recensione* “review” or *gradevole* “pleasant”) and emotional states (e.g. *ottimista* “optimist” and *attitudine* “aptitude”). The neighbors point to a meaning of *positivo* described in dictionaries[27] as usually employed in everyday language: "in an optimistic manner, with confidence, affirming the value of something or someone, good and favorable".

In the 2020 dataset, *positivo* has indeed narrowed its use to the medical semantic field: 8 out of its top-10 nearest neighbors are clearly connected with medicine and the pandemic. *tampone* (“swab”), *positività* (“positivity”), *40ena* (an abbreviation of *quarantena*, “quarantine”), *contagiare* (“to infect”), *sintomatico* (“symptomatic”), *asintomatico* (“asymptomatic”), [test] *sierologico* (“antibodies test”) and *infetto* (“infected”) all indicate a meaning of *positivo* as pertaining to medicine: a diagnostic response that confirms

[27] The definitions of word senses in this section are taken from the online dictionary Treccani, https://www.treccani.it/.

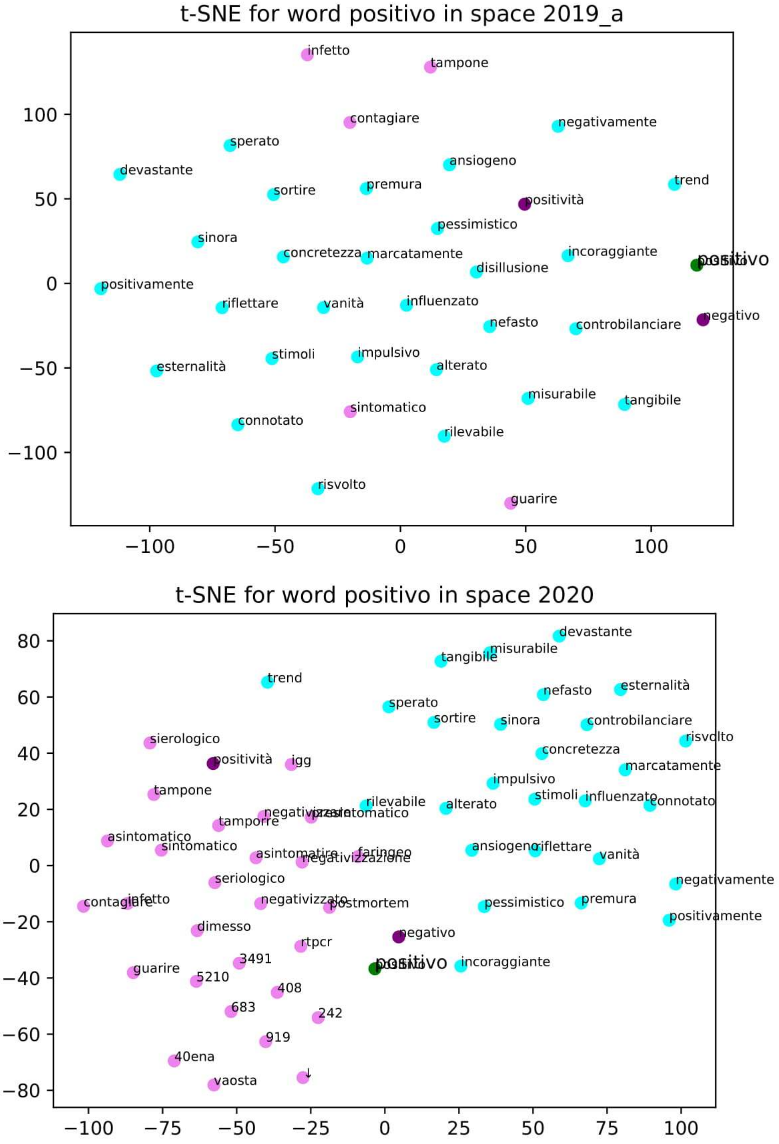


Figure 1
Visualization of *positivo* in the two sub-corpora

the formulated hypothesis, unfavorable to the tested subject, who, by extension, is also called *positivo*.

The other two nearest neighbors (NN) of *positivo*, *vaosta* and ↓, are less interpretable, however, can still be connected to the pandemic: the former is a shortening of Valle d'Aosta ("Aosta Valley"), the smallest Italian region, bordering France. *vaosta* occurs with the names of other regions in the daily tables listing COVID-19 cases and deaths. As to why only *vaosta* figures as a neighbor of *positivo* there is no evident clue. The symbol ↓ is present for the same reason: it occurs frequently in the periodic pandemic reports. The connection with the tallies of the pandemic is confirmed by the presence of numbers in the neighborhood of *positivo*.

Figure 1 gives the visual representation of the top-30 nearest neighbors of *positivo* in the two spaces as an example of the plots created by the detection algorithm. Cyan is used for the 2019 neighbors, while pink is used for the 2020 ones. Shared neighbors are marked in purple. As can be seen for the visual representation, the only neighbors in common between the two sub-corpora are *positività* and *negativo*, marked in purple in the plot. The collocations of these two words in the 2019 sub-corpus reveal that they are almost never used in a medical sense.

The usage of *intensivo* is also characterized by a narrowing. In the first dataset, *intensivo* occurs in expressions like *corso intentensivo* ("crash course") or *allevamento intensivo* ("intensive animal farming"). Its neighbors, however, are diverse: *vegetale* ("plant, plant-related") is used both in talking about agriculture and food. *integratore* ("nutritional supplement") and *proteina* ("protein") are found in sentences about *allenamento intensivo* ("intensive training"). *cbd* (cannabidiol), *thc* (tetrahydrocannabinol),[28] *molecola* ("molecule") and *nicotina* ("nicotine") are related to drugs. These could well be connected to *combustione* ("combustion"). The connection with *intestino* ("intestine") and *hiv* (human immunodeficiency viruses) are less clear. However, after looking at the 2020 sub-corpus, it is clear that the usage of *intensivo* in the 2019 dataset was more general.

Indeed, in the 2020 sub-corpus, the word *intensivo* is used more frequently, and all its neighbors belong to the healthcare vocabulary. *icu* (intensive care unit) and its Italian counterpart *rianimazione* ("reanimation", used interchangeably with *terapia intensiva*) are the firsts on the list. Further down, there are *ricovero/ospedalizzazione* ("hospitalization"), *ricoverare/ospedalizzare* ("to hospitalize"), *ricoverato/ospedalizzato* ("hospitalized"), and *ospedale* ("hospital"). *intubare* refers to the operation performed by doctors to install a breathing tube into the throat of a patient. Here the connection to the pandemic is pervasive, with some neighbors hinting at the strenuous conditions of hospitals during the pandemic (*saturato* "saturated"), and other names of Italian regions (*vaosta*,[29] *friulivg* (Friuli-Venezia Giulia)).

The usage of *guarire* ("to heal") also underwent narrowing, from a broader use in the medical semantic field, to a more restricted use regarding intensive care. In the 2019 space, the neighbors of *guarire* feature words such as *psicoterapeuta* ("psychotherapist"), *deprimere*, ("to depress") and *psicologo* ("psychologist") which pertain to mental health and therapy. *astinenza* ("abstinence", or "withdrawal") and *malessere* ("discomfort") can also be connected to this scope. *chirurgo* ("surgeon"), *dottoressa* ("female doctor") and *prescrivere* ("to prescribe") are more general. *intestino* and *involontariamente* ("unintentionally") seem incidental and are not particularly informative.

[28] CBD and THC are two of the cannabinoids found in cannabis.

[29] A wrongly lemmatized **vaosto* is present

Table 2
Neighbors of top-10 words. The upper line lists top-10 neighbors in the 2019 sub-corpus, while the lower line lists neighbors in the 2020 sub-corpus. Lemmatization errors, which are almost always easily understandable by a native speaker, are marked with a star (*) symbol.

	Neighbors
positivo	desiderato, decisivo, ottimista, lato, recensione, gradevole, normalità, attitudine, scaturire, deleterio tampone, positività, 40ena, contagiare, sintomatico, vaosta, asintomatico, ↓, sierologico, infetto
virus	vulnerabilità, bios, resettare, bug, chiavetta, terminale, criptare, diabete, cancro, scansione sarscov2, coronavirus, covid19, covid, ebola, contagio, patogeno, infettare, sars, aerosol
intensivo	vegetale, integratore, combustione, cbd, thc, molecola, nicotina, proteina, intestino, hiv icu, rianimazione, ospedalizzato, ricovero, ricoverare, ricoverato, ospedalizzare, intubare, ospedale, ospedalizzazione
testare	arduino, centralina, plugin, usato, apportare, alterare, simulare, falla, fungere, lsd tampone, sintomatico, testato, asintomatico, ospedalizzare, ct, diagnostico, tamponare, screening, infetto
guarire	psicoterapeuta, astinenza, prescrivere, chirurgo, dottoressa, intestino, deprimere, malessere, psicologo, involontariamente guarito, ospedalizzare, asintomatico, guarigione, clinicamente, ricoverato, contagiare, decesso, decedere, ospedalizzato
bla	professorone, la', sminuire, inerzia, passere, *ripartare, umiltà, contrattuale, pigrizia, *dirtelare superfluo, trito, figliare, *diciamocelare, egoistico, famigliare, etc, *smettilare, moralista, stufo
eco	intitolare, convegno, studentesco, seguace, sostenitore, invocare, *buongiorne, vocabolario, *altaforte, rivista asimov, pascolo, collana, lovecraft, mattone, divulgativo, *murakamo, microscopico, orwell, philip
leve	*capacitare, sara, immortale, magnetico, aggrappare, rivoluzionario, cosmico, *vadere, lentezza, nano inculare, perverso, rum, ftw, dinamico, trucchetto, proletariato, arrampicato, toppa, nervo
influenza	interferire, migratore, venezuelano, decisivo, presidenziale, caratterizzare, coinvolgimento, connotazione, competitività, oppressione influenzale, *polmonito, stagionale, sintomatologia, *polmonita, complicanza, ebola, mers, morbillo, contagiosità
gene	amministrare, neurone, tribù, bravura, portatore, risaputo, composto, azionista, squilibrio, prole dna, nomea, ariano, innato, land, sarscov2, mutazione, ceppo, cromosoma, mediocrità

In the 2020 sub-corpus, the neighborhood of *guarire* is focused on hospital and intensive care: *ospedalizzare*, *ospedalizzato*, *ricoverato*, *contagiare*, and *asintomatico* return as neighbors. These words are common also in the surroundings of other pandemic-related words. Other terms are both positive, like *guarito* ("healed") and *guarigione* ("healing"), and negative, like *decesso* ("death") and *decedere* ("to die"). These last two are commonly used in a more formal setting, like news, so are likely connected to *guarire* because of the reports on cases, deaths, and recoveries from COVID-19. This is confirmed by the presence of numbers in the larger neighborhood. *clinicamente* ("clinically") is more neutral, but still connected to the field of medicine. The only common neighbor between the two spaces is *guarigione*.

gene is the last detected word in the top-10 candidates which could be ascribed to the *narrowing* class. The low number of occurrences does not allow for a coherent embedding representation in both spaces, even if the 2020 one seems slightly better than the one from the previous year's dataset. While the neighborhood in the 2019 space is quite diverse, in the 2020 one can identify some connections: *dna*, *ariano* ("arian"), *mutazione* ("mutation"), *ceppo* ("strain", meaning a variant of e.g. a virus), *cromosoma* ("chromosome") relate to the genetic sense of *gene*. *nomea* ("reputation"), *innato* ("innate"), and by contrast *mediocrità* ("mediocrity"), loosely hint at sense of "genius". However, the influence of the pandemic may have prompted more discussion that involved the biological gene: *sarscov2* is listed as a neighbor in the 2020 space. Interestingly, there are no shared words in the neighborhoods for the two spaces.

The usage of *virus* shifted from informatics to the pandemic. Eight out of ten neighbors in 2019 point to the computer version of a virus: *vulnerabilità* ("vulnerability" of a system), *bios*,[30] *resettare* ("to reset"), *bug*,[31] *chiavetta* ("USB pen-drive"), *terminale* ("terminal"), *criptare* ("to encrypt"), and *scansione* ("scan"). On the contrary, *diabete* ("diabetes") and *cancro* ("cancer") are medical terms.

In the 2020 sub-corpus, the connection with the pandemic is explicit in the first four neighbors, which are all variations of COVID-19: *sarscov2*, *coronavirus*, *covid19* and *covid*. The other neighbors are still correlated with specific diseases, like *ebola* and *sars*, and their spread, *contagio* ("contagion"), *patogeno* ("pathogen"), *infettare*. *aerosol* refers to the fact that COVID-19 is spread by droplets of saliva in the air. Even looking at a larger list of neighbors, there are no words related to the computer sense of *virus* during 2020. Common neighbors between the two corpora are *infettare*, *contagioso*, *infezione*, *hiv*.

Also the usage of *testare* has shifted from practical engineering to medical testing. Some neighbors of *testare* ("to test") in the 2019 sub-corpus hint at electronic and digital devices: *arduino*,[32] *centralina* ("control unit"), *plugin*, all belong to this semantic field. Other neighbors include the verbs *alterare* ("to alter"), *simulare* ("to simulate") and *fungere* ("to function"), which are used in diverse situations. *usato* ("used"),*falla* ("fault") and *lsd* (lysergic acid diethylamide) range from generic to very specific. As it was the case with *intensivo*, again the sparse use of this word in the 2019 dataset renders its representation imprecise, overly influenced by usage in only certain limited discussions.

[30] Acronym for Basic Input/Output System, a firmware (a special kind of software that provides low-level control for a specific hardware) used to perform hardware initialization during the power-on startup process, known as booting.

[31] A software bug is an error, flaw, or fault in a computer program that causes an incorrect or unexpected result or behavior. This is indeed the sense of bug here, as it is common to use English loanwords or calques in Italian for the digital semantic field in general.

[32] Arduino is an open-source hardware and software company project and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices.

In the 2020 sub-corpus, *testare* is well represented and connected to the pandemic: *tamponare* arises as a specialized version of *testare*, with the specific meaning of "to test with a swab (*tampone*)". This is connected to *screening*, a loanword that refers to medical testing. The acronym *ct* is used interchangeably for COVID-19 test and *commissario tecnico* ("sports coach"). *diagnostico* ("diagnostic") seems to be used in connection with *tampone* and other testing methods and equipment, confirming the connection to the pandemic. *sintomatico* ("symptomatic"), *asintomatico* ("asymptomatic"), *infetto*, and *ospedalizzare* are clearly connected to the virus.

tamponare is present in the list also as a neighbor of *isolare*, which is discussed in Section 4.2. This is an interesting case: this meaning of *tamponare* as "to perform a swab" is listed as a 2020 neologism derived from *tampone* ("swab") on the on-line version of the Treccani dictionary. This is supported by the data in our corpus: its nearest neighbors in the 2020 space are words such as *testare*, *sintomatico*, *malauguratamente* ("unfortunately"), *tampone*, *ospedalizzare*, and *profilassi* ("prophylaxis"). Thus, this lemma is an homonym of *tamponare* in the sense of "to hit, with the anterior part of a vehicle, the back of another vehicle in the same lane". The neighbors of *tamponare* in the 2019, point only at this sense: *cid*, abbreviation for "Convenzione d'Indennizzo Diretto, or Constatazione Amichevole d'Incidente Stradale" (lit. "Direct Compensation Convention" or "Friendly Verification of Car Accident"),[33] clearly pertains to the car accident situation. Other neighbors in 2019 are *bagagliaio* ("trunk"), *sopraggiungere* ("to arrive, usually suddenly and unexpectedly"), *retromarcia* ("reverse [gear]"), *frenata* ("hard braking"), *frenare* ("to brake"), *conducente* ("driver"), *semaforo* ("traffic light").

The word *influenza* has shifted its use, too. Its neighbors in the 2019 sub-corpus point at the sense of "action done by one thing or person on another one": *interferire* ("to interfere"), *coinvolgimento* ("involvement"), and *caratterizzare* ("to characterize") are quite indicative in this sense. Some other hint at a more geo-political use of the same sense of *influenza*: *presidenziale* ("presidential"), *venezuelano* ("Venezuelan"), *oppressione* ("oppression"), and **migratore* (maybe *migratoria* "migratory", as in *flussi migratori*). Collocations shows that *presidenziale* is used referring to American politics, while *venezuelano* refers to the Venezuelan crisis in the beginning of 2019. *connotazione* ("connotation"), *competitività* ("competitivity") and *decisivo* ("decisive") are the remaining neighbors.

In the 2020 sub-corpus, *influenza* shifts completely to a medical usage: some of its neighbors refer to diseases, such as *ebola*, *mers*, *morbillo*, **polmonito* and **polmonita* (correct form: *polmonite*, "pneumonia"). *stagionale* is coming from *influenza stagionale* ("common flu"), while *influenzale* ("flu-related"), *sintomatologia* ("symptomatology"), *complicanza* ("complication"), and *contagiosità* ("the ability or state to be infective") relate to the effect of *influenza*. Even in the larger neighborhood there is no trace of the usages attested in 2019. Moreover, there are no neighbors in common between the two datasets.

bla is the first of the three candidates which neighbors are not informative with respect to usage change. *bla*, usually repeated two or three times (*bla bla*), is a common onomatopoeia indicating useless conversations or futile chatter. The frequency of this word grew only slightly, in line with the overall increment in the size of the data. In both spaces the neighbors of *bla* are not informative. The only ones that can be somewhat connected with the common use of *bla* are found in the 2020 space: *trito* ("crushed"), can be used in the idiomatic expression *trito e ritrito* ("grounded and grounded again") meaning something that is used or said too much, commonly known, prosaic and trivial.

[33] This is referring both to a procedure and its related form that allows for more smooth insurance compensation of the damage.

Table 3
Absolute and relative frequencies of top-10 words. Relative frequencies are calculated as the number of occurrences of a word divided by the total number of tokens in the lemmatized corpus for a specific year.

	Absolute Frequency		**Relative Frequency (%)**		**Increase (%)**
	2019	**2020**	**2019**	**2020**	
positivo	2969	13688	0.52169	1.78806	1.26637
virus	270	16632	0.04744	2.17263	2.12519
intensivo	203	3716	0.03567	0.48542	0.44975
testare	439	3347	0.07714	0.43722	0.36008
guarire	238	2454	0.04182	0.32057	0.27875
bla	302	489	0.05307	0.06388	0.01081
eco	256	436	0.04498	0.05695	0.01197
leve	269	369	0.04727	0.04820	0.00093
influenza	996	4226	0.17501	0.55204	0.37703
gene	354	524	0.06220	0.06845	0.00625

Others are *superfluo* ("excessive") and *etc* (abbreviation of "etcetera"). The neighbors show some lemmatization issues: **ripartare* instead of *ripartire* ("to start again"), **dirtelare* most certainly derived from *dirtelo* ("to say to you", *-lo* is an enclitic second person pronoun), **diciamocelare* from *diciamocelo* ("to say to ourselves", often said with the sense of "let's be clear to/real with ourseves") and **smettilare* from *smettila* ("stop it!").

Also not very informative are the neighbors of *eco*, which can indeed be the common noun for "echo", a reflection of sound; or, if one looks at its nearest neighbors, the famous writer Umberto Eco, at least in the 2020 sub-corpus. However, in the 2019 dataset the neighborhood is less clear even if they are clearly related to literature: for example, *pascolo* referring to poet Giovanni Pascoli and *murakamo* referring to Japanese writer Murakami Ryū. Other words connected with the literary world are *collana*, a series of books, *mattone*, a long and tedious book (lit. "a brick"), and *divulgativo*, usually a science or otherwise academic book intended for the general audience. *philiph* is referring to sci-fi author Philip K. Dick, as suggested by the presence of other writers of the same genre like Isaac Asimov (*asimov*) and George Orwell (*orwell*).

leve is maybe the least informative entry in this list: it is a lemmatization error, since the lemmatizer did not use the citation form, the singular *leva* ("lever"). In addition, in some cases *leve* can derive from the name of Holocaust survivor and writer Primo Levi. The neighborhood of *leve* in both corpora emerges from limited interactions in peculiar discussions: just to cite one, the first neighbor of the 2020 list, a profanity that literally means "to sodomize", is due to an exchange between two users on day 261 of 2020, where the word *leve* was used about 20 times.

The presence of these three cases, *eco*, *bla*, *leve*, can be attributed to their inaccurate embedding representations, which are in turn due to their scarce frequency. Their word embeddings are overly influenced by some peculiar context of use, which renders their neighborhoods less informative to define their usage. Frequency-wise all three share a pattern of just a slight increase, in line with the growth of the data for the second corpus. A preliminary analysis of the other results in the top-100 detected words shows that this can be the case for many other relatively low-frequency words.

Overall, in these top-10 candidates there are 6 informative results (*positivo*, *virus*, *intensivo*, *testare*, *guarire*, *influenza*, *gene*), and 3 less informative results (*bla*, *eco*, *leve*). These last three candidates have imprecise embedding representations, due to their low frequency of use. Also, wrong lemmatization may have had an impact. However, among all the 200 neighbors of the words listed in the top-10, just 11 are wrongly lemmatized, and even in these cases the errors are intelligible for a native speaker.

Sometimes, the embedding representations of the top-10 candidates show some less specific neighbors, at least in the 2019 space, where their frequency is lower. It is interesting to note, however, that all the informative candidates had a noticeable growth in relative frequency in the 2020 sub-corpus. As seen with informative outputs, if the changed word is well represented, it is also detected by the algorithm. Table 3 gives frequency data for the top-10 candidate words proposed by the algorithm.

Even if the top-10 results are not totally devoid of problems, the output for an unlemmatized corpus seems worse, with only five terms (*virus*, *bla*, *vaccino*, *positivo*, and *positivi*) having some significant increase in frequency. As seen in the previously discussed words, low frequency leads to imprecise representations built only on a handful of particular discussions. This naturally leads to radically different neighborhoods over the two corpora, tricking the algorithm into thinking that these cases are instances of usage change. It can be argued that these words have in fact undergone usage change, that is, they have changed contexts of use, but their neighbors often give no clue to their meaning, calling into question their validity. While far from being perfect, lemmatization seems to smooth out at least some of these cases. Pertaining to specific results in the unlemmatized top-10, *bla* presents the same problems as explained above; *positivo* and *positivi* are two inflected forms of the same lemma (thus of low informative value when searching for changes in language use), but overall correctly labeled as changed; *peste* and *vaccino* have a clear enough representation only in the second sub-corpus; *fico* has sparse usage in both datasets, but it is clear only in the first one. *capitano* is the only case with decreased frequency: in the 2019 space it is related almost exclusively with the discussion around the incident involving NGO ship captain Carola Rackete and her antagonist, then Interior Minister Matteo Salvini, sometimes nicknamed "Il Capitano". In the 2020 space many neighbors point to *càpitano* as a verb ("they happened", as opposed to *capitàno*, "captain").

As for the top-10s detected with the method based on the alignment of the vector spaces (AlignCos, Hamilton, Leskovec, and Jurafsky (2016)), in both cases they are much worse than those found by the Nearest Neighbors method (Gonen et al. 2020): in the lemmatized version, only *intensivo* is significative, all the others being cases of representations skewed by low frequency. In the unlemmatized version two candidates are somewhat valid: *fontana* changed its use from "fountain" in the 2019 space to referring to Attilio Fontana, the governor of Lombardy, the Italian region hit the worst by the pandemic. The other significant candidate is *vaccino*, which has a low frequency in 2019 and an obviously good representation in 2020.

Table 4
Neighbors of other relevant words in the top-100. The upper line lists top-10 neighbors in the 2019 sub-corpus, while the lower line lists neighbors in the 2020 sub-corpus. Lemmatization errors, which are almost always easily understandable by a native speaker, are marked with a star (*) symbol.

	Neighbors
vaccino	omeopatico, prescrivere, biologo, dermatologo, ricoverare, cancro, esente, omosessualità, *asile, prescrizione antinfluenzale, antivirale, pfizer, morbillo, vaccinato, oxford, anticorpo, somministrare, cavia, gregge
riaprire	avviare, portone, archiviare, spostato, *accendare, riprovare, quirinale, rimbalzare, sbucare, avvio riapertura, richiudere, ripartenza, palestre, maggio, allentare, restrizione, *asile, allentamento, *contage
tappeto	muffa, vernice, soffitto, siringa, lavandino, cemento, ombrellone, rame, tavoletta, pallino testare, tampone, sierologico, screening, capillare, prericovero, isolare, quarantene, molecolare, test
normalità	maschilista, retrogrado, omofobia, immaturo, socialmente, bigotto, ansioso, omosessualità, geloso, deleterio riaprire, andata, allentamento, parvenza, gradualmente, autunno, riapertura, intimità, ricaduta, esodo
morto	persecuzione, attenuante, portatore, perseguitare, ignoto, terrorizzare, rapire, concentramento, stupratore, molestia ospedalizzato, decesso, contagiato, infettato, ricoverato, *contage, ospedalizzare, decedere, *muoiare, contagiare
curva	ultras, tifoso, tifoseria, vicolo, tir, marce, interista, lazio, hamilton, *filmetro esponenziale, appiattire, curvo, contage, accelerazione, pendenza, impennare, picco, r0, progressione
isolare	inadatto, interferire, sabotare, dovunque, volente, fomentare, deviare, bollare, nolente, quotidianità circoscrivere, *focolao, quarantenare, sintomatico, blindare, rintracciare, *diffondare, asintomatico, tamponare, contagio
ondata	antisemitismo, migratore, sovranismo, generazionale, retorica, buonismo, berlusconiano, reazionario, consumismo, apice epidemia, *focolao, impennata, lockdown, esodo, autunno, scongiurare, riapertura, pandemia, *tsuname
terapia	omeopatico, prevenzione, malessere, allergia, raffreddore, relazionale, stigma, erezione, molecola, ansioso rianimazione, ricovero, ospedalizzare, intubare, icu, ospedalizzato, farmacologico, ospedalizzazione, ricoverato, ormonale
rosso	guancia, muffa, romeo, cera, hamilton, mela, dannato, illuminare, adesivo, cappello zona, grigie, nembro, lodigiano, tonalità, *mantovo, tartufo, codogno, stemma, *cremono
scorta	saviano, divisa, domiciliare, proiettile, equipaggio, rinchiudere, digos, immunità, rimozione, vendicare rifornire, scarseggiare, *vivero, igienizzante, *amuchino, introvabile, assaltare, monouso, sottovuoto, ricambi
chiuso	finito, tappo, elefante, fogna, circolo, serratura, sbraitare, sfortunatamente, *tienire, cassetto blindare, palestre, richiudere, affollare, riapertura, blindato, battente, scappato, ammassare, confinare
fontana	basilica, sant', cimitero, sottosegretario, virginia, passeggiare, *lucco, riva, villa, portone *gallero, cirio, *zaio, gallera, *bonaccino, *formigone, assessore, *camico, *umilansifonsifere, lombardia
emergenza	abitativo, naufrago, irregolarità, *rimpatro, rifugiare, irregolare, incidente, interruzione, *lampeduso, generatore pandemia, emergenziale, pandemico, epidemia, fronteggiare, proroga, imprevista, commissariamento, ripartenza, *covere
paziente	lucidità, malessere, dolore, nutrizionista, ossessivo, allergia, seduta, ansioso, erezione, perizia rianimazione, ospedalizzare, intubare, 38enne, anestesista, complicanza, *polmonita, oncologico, ricoverato, diagnosticato
malato	frustrato, sindrome, isterico, risvegliare, disordine, omeopatico, ossessivo, immaturo, retrogrado, daenerys oncologico, ammalato, infetto, ospedalizzare, contagiato, diagnosticato, immunodepresso, *ammalere, ricoverato, contagiare

4.2 Other relevant words

Other words in the top-100 candidates for usage change are relevant. These and their neighbors are listed in Table 4 and are briefly discussed below.

Notable cases of usage narrowing include *vaccino*, *terapia* (“therapy”), *malato* (“ill”), and *paziente* (“patient”). The neighbors of these words change from generally mild connotation (e.g. *terapia* is used in the 2019 sub-corpus with *omeopatico* “homeopathy”, *allergia* “allergy”, and *raffreddore* “common cold”), to a more severe one, related to the pandemic (e.g. *terapia* is used in the 2020 sub-corpus with *rianimazione*, *intubare*, and *icu*).

Other instances of narrowing, or change to a specific usage, are those of *curva* (“curve“), *rosso* (“red“), and *isolare* (“to isolate“), which becomes to be specifically connected to the pandemic. Among the latter’s neighbors *quarantenare* (“to quarantine“) is found. The case of *quarantenare* is interesting, despite very low occurrences. This verb is found a dozen of times in the 2020 space, but only one in the 2019 one. It is also not present in the Treccani dictionary, not even as a neologism. *quarantenare* is used in the sense of “to quarantine“. The 2019 instance refers to the process with which Reddit administrators hide and close a subreddit deemed to be harmful or not in line with the platform’s rules. The sense in 2020 is similar, but the term is used always in connection with the pandemic: neighbors include *quarantena* (lemmatized as **quaranteno*), *fiduciario* (“fiduciary“, in the more formal expression “quarantena fiduciaria“), *infettare* (“to infect“), *autoisolamento* (“self-isolation“), *isolamento* (“isolation“), *precauzionale* (“precautionary“).

The neighbors of *normalità* (“normality“), *tappeto* (“carpet“), *morto* (“dead“), *ondata* (“wave“), *scorta* (both “security detail” and “stockpile”), *emergency* (“emergenza“) point to shifts in usage. *normalità* shift its usage from gender issues to the pandemic; *tappeto* moves from homes to “carpet testing“; *morto*’s usage chages from crimes to pandemic deaths; *ondata* from a geopolitical usage to describing the successive waves of the disease; *scorta* switches from “security detail“ to “stockpile“; and *emergenza* refocuses from the migrants crisis to the pandemic.

Table 5 gives frequency data on other relevant words in the top-100, The words are ordered according to their position in the list proposed by the algorithm (not always contiguous), as already seen for the top-10 candidates; words which have indeed experienced change have also a noticeable increase in relative frequency, albeit less than for the terms in the top-10.

5 Conclusions

This work started from the hypothesis that a global crisis like the COVID-19 pandemic could impact language use. This was verified with computational means, leveraging both theoretical linguistics and NLP techniques. A corpus was created by scraping online text from the Italian Reddit community. The data was collected for the days between January 30 and November 30 of both 2020 and 2019, creating two sub-corpora. The raw text was then cleaned and lemmatized to allow further analysis. This dataset alone, both raw and preprocessed, could be a useful resource for other applications and it is publicly available. Future work may focus on the extension of the dataset’s timeframe.

This research follows previous work and methodology in the field of computational language change detection, focusing on short-term usage change. To the best of our knowledge, this is the first work of this kind done for Italian, both in the field of

Table 5
Absolute and relative frequencies of other relevant words in the top-100. Relative frequencies are calculated as the number of occurrences of a word divided by the total number of tokens in the lemmatized corpus for a specific year.

	Absolute Frequency		Relative Frequency (%)		Increase (%)
	2019	2020	2019	2020	
vaccino	710	5677	0.12476	0.74158	0.61683
riaprire	311	3775	0.05465	0.49313	0.43848
tappeto	226	787	0.03971	0.10281	0.06309
normalità	265	1185	0.04656	0.15480	0.10823
morto	2201	7426	0.38675	0.97006	0.58331
curva	692	1670	0.12159	0.21815	0.09656
isolare	240	994	0.04217	0.12985	0.08767
ondata	233	1750	0.04094	0.22860	0.18766
terapia	738	4857	0.12968	0.63447	0.50479
rosso	2791	6457	0.49042	0.84348	0.35306
scorta	441	1047	0.07750	0.13677	0.05928
chiuso	1856	7148	0.32613	0.93374	0.60762
fontana	253	1264	0.04446	0.16512	0.12066
emergenza	1131	5399	0.19873	0.70527	0.50654
paziente	873	4125	0.15340	0.53885	0.38545
malato	954	3289	0.16763	0.42964	0.26201

COVID-19-related linguistic research and short-term language change detection. The latter is carried out with the neighborhood-based method outlined in Gonen et al. (2020), previously untested for the Italian language. The choice to lemmatize the data allowed to evaluate the impact of this pre-processing step on the method.

The initial research questions were the following: has the pandemic impacted the usage of the Italian language? Can this impact be detected with computational means? In fact, the manual analysis of the results produced by the algorithm showed that, as expected, some degree of usage change has occurred. The computational method used to detect it has shown to be quite solid also for Italian. Our experiments have shown that lemmatization as a pre-processing phase is important for Italian, given that without this step the results were less informative, although it remains a challenging task for an inflectional language such as Italian. Future work may involve an improved lemmatization.

A similar work for English by Guo, Xypolopoulos, and Vazirgiannis (2022) adopts different methodological choices, such as the alignment approach of Hamilton, Leskovec, and Jurafsky (2016) to detect usage change, and the analysis of a selection of predefined keywords. Its results are therefore not comparable to ours. Nevertheless, it shows that, also for English, a shift in usage is detected towards COVID19 and healthcare related words.

It remains to be seen if the change in usage will translate in actual lasting mutations in the language. The rise of of a new word as in the case of *tamponare* "to perform a swab", may be more significant and enduring than the already existing, but domain-specific sense of *positivo* "a diagnostic response that confirms the formulated hypothesis, unfavorable to the tested subject", which became widespread due the pandemic. These cases seem more typical of short-term usage change: more specific, or different senses of a word increase their use, and overtake the more established senses due to a plethora of factors, in this case the pandemic. The surge of these senses may well be temporary.

In conclusion, this work successfully contributed to: (i) the creation of a new dataset, focusing on short-term usage change for Italian; (ii) the cross-linguistic application of a relatively novel method of language change detection; (iii) a linguistic analysis of the impact of the pandemic on language use in a language other than English. All the data and part of the code created for this work are publicly available online.[34]

Acknowledgments

We would like to thank the reviewers for their valuable comments and suggestions.[35]

References

Bamler, Robert and Stephan Mandt. 2017. Dynamic word embeddings.

Basile, Pierpaolo, Annalina Caputo, Tommaso Caselli, Pierluigi Cassotti, and Rossella Varvara. 2020a. A Diachronic Italian Corpus based on “L’Unità”. In *Proceedings of the 7th Italian Conference on Computational Linguistics (CLiC-it 2020)*, volume 2769, Online, March 1-3, 2021. Italian Association for Computational Linguistics.

Basile, Pierpaolo, Annalina Caputo, Tommaso Caselli, Pierluigi Cassotti, and Rossella Varvara. 2020b. DIACR-Ita @ EVALITA2020: Overview of the EVALITA2020 Diachronic Lexical Semantics (DIACR-Ita) Task. In *Proceedings of the Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*, Online, December.

Basile, Pierpaolo, Annalina Caputo, Roberta Luisi, and Giovanni Semeraro. 2016. Diachronic analysis of the italian language exploiting google ngram. In Pierpaolo Basile, Anna Corazza, Francesco Cutugno, Simonetta Montemagni, Malvina Nissim, Viviana Patti, Giovanni Semeraro, and Rachele Sprugnoli, editors, *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)*, volume 1749 of *CEUR Workshop Proceedings*, Napoli, Italy, December 5-7. CEUR-WS.org.

Basile, Pierpaolo, Annalina Caputo, and Giovanni Semeraro. 2014. An enhanced Lesk word sense disambiguation algorithm through a distributional semantic model. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1591–1600, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.

Blank, Andreas. 1999. Why do new meanings occur? A cognitive typology of the motivations for lexical semantic change. *Historical Semantics and Cognition*.

Cafagna, Michele, Lorenzo De Mattei, and Malvina Nissim. 2020. Embeddings-based detection of word use variation in italian newspapers. *Italian Journal of Computational Linguistics*, 6:9–22, 12.

Del Tredici, Marco and Raquel Fernández. 2018. The road to success: Assessing the fate of linguistic innovations in online communities. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1591–1603, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Del Tredici, Marco, Raquel Fernández, and Gemma Boleda. 2019. Short-term meaning shift: A distributional exploration. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2069–2075, Minneapolis, Minnesota, June. Association for Computational Linguistics.

[34] https://github.com/edoardosignoroni/usage_change_ITA

[35] The paper is a re-elaboration of Edoardo Signoroni’s Master Thesis defended at the University of Pavia. For the only purposes of the Italian Academia, Elisabetta Jezek is responsible for sections 1 and 2, and Edoardo Signoroni for sections 3, 4 and 5. Rachele Sprugnoli edited the paper before the submission.

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

Eger, Steffen and Alexander Mehler. 2016. On the linearity of semantic change: Investigating meaning variation via dynamic graph models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 52–58, Berlin, Germany, August. Association for Computational Linguistics.

Gonen, Hila, Ganesh Jawahar, Djamé Seddah, and Yoav Goldberg. 2020. Simple, interpretable and stable method for detecting words with usage change across corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 538–555, Online, July. Association for Computational Linguistics.

Gulordava, Kristina and Marco Baroni. 2011. A distributional similarity approach to the detection of semantic change in the Google Books ngram corpus. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 67–71, Edinburgh, UK, July. Association for Computational Linguistics.

Guo, Yanzhu, Christos Xypolopoulos, and Michalis Vazirgiannis. 2022. How COVID-19 is Changing Our Language: Detecting Semantic Shift in Twitter Word Embeddings. In *Conférence Nationale en Intelligence Artificielle 2022 (CNIA 2022)*, Actes CNIA 2022, Saint-Etienne, France, June.

Hamilton, William L., Jure Leskovec, and Dan Jurafsky. 2016. Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1501, Berlin, Germany, August. Association for Computational Linguistics.

Harris, Zellig. 1954. Distributional structure. *WORD*, 10(2-3):146–162.

Ježek, Elisabetta. 2016. *The Lexicon, an Introduction*. Oxford Textbooks in Linguistics. Oxford University Press, Oxford.

Jurafsky, Dan and James H. Martin. 2021. *Speech and Natural Language Processing (3rd ed. draft)*. Pearson Prentice Hall. retrieved from https://web.stanford.edu/ jurafsky/slp3/.

Kahmann, Christian, Andreas Niekler, and Gerhard Heyer. 2017. Detecting and assessing contextual change in diachronic text documents using context volatility. In *Proceedings of the 9th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, pages 135–143, Funchal, Madeira, Portugal, November.

Kaiser, Jens, Dominik Schlechtweg, and Sabine Schulte im Walde. 2020. OP-IMS @ DIACR-Ita: Back to the Roots: SGNS+OP+CD still rocks Semantic Change Detection. In *Proceedings of the Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*, Online, December.

Kim, Yoon, Yi-I Chiu, Kentaro Hanaki, Darshan Hegde, and Slav Petrov. 2014. Temporal analysis of language through neural language models. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pages 61–65, Baltimore, MD, USA, June. Association for Computational Linguistics.

Kulkarni, Vivek, Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2014. Statistically significant detection of linguistic change.

Kutuzov, Andrey, Lilja Øvrelid, Terrence Szymanski, and Erik Velldal. 2018. Diachronic word embeddings and semantic shifts: a survey. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1384–1397, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Lenci, Alessandro. 2018. Distributional models of word meaning. *Annual Review of Linguistics*, 4(1):151–71.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Prazák, Ondrej, Pavel Pribán, and Stephen Taylor. 2020. UWB @ DIACR-Ita: Lexical Semantic Change Detection with CCA and Orthogonal Transformation. In *Proceedings of the Seventh Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2020)*, Online, December.

Qi, Peng, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Online, July.

Rodda, Martina A., Marco Senaldi, and Alessandro Lenci. 2017. Panta rei: Tracking Semantic Change with Distributional Semantics in Ancient Greek. *Italian Journal of Computational Linguistics*, 3:11–24, 06.

Rudolph, Maja and David Blei. 2018. Dynamic bernoulli embeddings for language evolution. In *Proceedings of The Web Conference 2018*, Lyon, France, April. ACM.

Sagi, Eyal, Stefan Kaufmann, and Brady Clark. 2009. Semantic density analysis: Comparing word meaning across time and phonetic space. In *Proceedings of the EACL 2009 Workshop on GEMS: Geometrical Models of Natural Language Semantics*, pages 104–111, Athens, Greece, March.

Sahlgren, Magnus. 2008. The distributional hypothesis. *Italian Journal of Linguistics*, 20(1):33–53.

Schlechtweg, Dominik, Barbara McGillivray, Simon Hengchen, Haim Dubossarsky, and Nina Tahmasebi. 2020. SemEval-2020 task 1: Unsupervised lexical semantic change detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1–23, Barcelona (online), December. International Committee for Computational Linguistics.

Stewart, Ian, Dustin Arendt, Eric Bell, and Svitlana Volkova. 2017. Measuring, predicting and visualizing short-term change in word representation and usage in vkontakte social network. In *Eleventh international AAAI conference on web and social media*, Montreal, Canada, March.

Tahmasebi, Nina, Lars Borin, and Adam Jatowt. 2021. Survey of computational approaches to lexical semantic change detection. *Computational approaches to semantic change*, pages 1–91.

Tang, Xuri, Weiguang Qu, and Xiaohe Chen. 2013. Semantic change computation: A successive approach. In Longbing Cao, Hiroshi Motoda, Jaideep Srivastava, Ee-Peng Lim, Irwin King, Philip S. Yu, Wolfgang Nejdl, Guandong Xu, Gang Li, and Ya Zhang, editors, *Behavior and Social Computing*, pages 68–81, Cham. Springer International Publishing.

Tang, Xuri, Weiguang Qu, and Xiaohe Chen. 2016. Semantic change computation: A successive approach. *World Wide Web*, 19.

Yao, Zijun, Yifan Sun, Weicong Ding, Nikhil Rao, and Hui Xiong. 2018. Dynamic word embeddings for evolving semantic discovery. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, Los Angeles, California, USA, February. ACM.

Extract Similarities from Syntactic Contexts: a Distributional Semantic Model Based on Syntactic Distance

Alessandro Maisto[*]
Università degli Studi di Salerno

Distributional Semantics (DS) models are based on the idea that two words which appear in similar contexts, i.e. similar neighborhoods, have similar meanings. This concept was originally presented by Harris in his Distributional Hypothesis (DH) (Harris 1954). Even though DH forms the basis of the majority of DS models, Harris states in later works that only syntactic analysis can allow for a more precise formulation of the neighborhoods involved: the arguments and the operators.

In this work, we present a DS model based on the concept of Syntactic Distance inspired by a study of Harris's theories concerning the syntactic-semantic interface. In our model, the context of each word is derived from its dependency network generated by a parser. With this strategy, the co-occurring terms of a target word are calculated on the basis of their syntactic relations, which are also preserved in the event of syntactical transformations. The model, named Syntactic Distance as Word Window (SD-W2), has been tested on three state-of-the-art tasks: Semantic Distance, Synonymy and Single Word Priming, and compared with other classical DS models. In addition, the model has been subjected to a new test based on Operator-Argument selection. Although the results obtained by SD-W2 do not always reach those of modern contextualized models, they are often above average and, in many cases, they are comparable with the result of GLOVE or BERT.

1. Introduction

Distributional Semantics (DS) is a model of meaning whose theoretical foundation is the Distributional Hypothesis (DH). DH relies on the work of Harris (Harris 1954), which sets out the basis for a linguistic distributional methodology. The Distributional Hypothesis states that the statistical distribution of linguistic elements in context determines their semantic behavior (Lenci 2018).

In Distributional Semantics, the similarity between two words is calculated in terms of similarity between vectors. Word vectors describe the terms as a numerical representation of the various contexts in which they appear. Lenci (2018) reported two kinds of classification for DS models: the first regards the type of context, the latter the method of learning distributional vectors. Regarding the first classification, we can identify *region models*, in which the context of a word is the entire region the word appears in, and *word models*, which calculate context as a set of terms that appear at a certain distance from a target word. With reference to the first family of models, Ruge (1992) claims that the

[*] Università degli Studi di Salerno, Via Giovanni Paolo II, 139, Fisciano (SA) Italia. E-mail: amaisto@unisa.it.

larger the context, the larger the number of not semantically compatible terms included in the analysis. Moreover, Sahlgren (2008) considered the document as a context for a legacy of information retrieval. Since information retrieval is an artificial problem, "a document in the sense of a topical unit–unity is an artificial notion that hardly exists elsewhere" (Sahlgren 2008).

Word models, on the other hand, can be further divided into *window-based models* and *syntactic models*: the former consider a variable number of neighbor terms (the so-called "window") as the context of a given word. The latter seek to exploit syntactic dependency in order to obtain a more precise simulation of human knowledge-learning phenomena. However, considering the amount of pre-processing required, there is no empirical evidence for the supremacy of this kind of model (Sahlgren 2008).

In this paper, we aim to investigate the benefits of using syntactic information in Distributional Semantics, regardless of the amount of pre-processing required (this is not really a problem because of advances in syntactic parsing and machine performance, as well as the availability of ever-larger parsed corpora). We present a new syntactic model that benefits from a deeper reading of Harris's theories. We have based the new model on the concept of *syntactic distance* (Liu, Xu, and Liang 2017), the distance between a target word and other words syntactically connected to it, calculated by a dependency parser (Definition 1).

Definition 1
The Syntactic Distance is equivalent to the number of arcs of the dependency graph which separate two words.

All words at a certain syntactic distance from the target word may be included in the context of the target word. We have named our model the *Syntactic Distance as Word-window* (SD-W2) to highlight its use of the syntactic distance as a context-window selection metric.

The Distributional Hypothesis stated by Harris includes a level of syntactic analysis, which our model incorporates by taking a parsed corpus as input. The preliminary results show that our dependency-based system achieves results that are comparable to many other models and very close to the results of the BERT-based models.

The paper is structured as follows: in section 2, we analyze Harris's studies on the concept of "distribution", exploring the reasons why a syntactic model must be implemented. In section 3 we present a brief state of the art and point the focus on the most related works. In section 4, we present our methodology. In section 5 we present the experimental step. Finally, in section 6 we present the experiment outline and results.

2. The Distributional Hypothesis

Harris (1954) claimed that when someone speaks, they choose the next word from the members of those classes of words that usually occur in this position. Each language element can be grouped into classes, and while the relative occurrence of a class can be stated exactly, the occurrence of a particular member of one class relative to a particular member of another class must be calculated in terms of probability. In other words, given two linguistic elements A and B, if they "have almost identical environments", they can be considered synonyms (e.g., *oculist* and *eye-doctor*); if they have "some environments in common and some not" (e.g., *oculist* and *lawyer*), they have different meanings and this difference corresponds to the "amount of difference of their environment" (Harris 1954, p. 157). The distributional structure reflects a sort of meaning structure in the way that "difference of meaning correlates with the difference of distribution" (Harris 1954, p. 156). The operation that studies the distributional structure is distributional analysis.

Distributional analysis is a basic process that Harris describes as being related to five distributional facts: a) possibility of *segmenting flows of speech* into parts (elements) to find regularities in the occurrence of one part relative to others; b) *similarity*, considered as the property of some elements to group with similar elements into sets; c) *dependence* of the elements in a group of similar objects on elements in another group; d) *substitutability* of elements that have the same environment; e) *domain*, such as the word, the phrase, the clause, in which both dependence and substitutability work. The distributional analysis output is a set of substitution classes or equivalence classes (Harris 1946, 1952).

Many authors have adopted the distributional hypothesis and the correlation between distribution and meaning for practical tasks: the first authors to exploit distributional analysis in a computational task were Schutze and colleagues (Schutze 1992a; Schütze 1992b; Schutze and Pedersen 1995). He presented a paper on word sense disambiguation based on a vector representation of word similarity derived from lexical co-occurrence. Subsequently, Landauer and Dumais (1997) proposed a model for the simulation of knowledge-learning phenomena based on local co-occurrence data in a large representative corpus, called Latent Semantic Analysis (LSA). Lund and Burgess (1996) introduced Hyperspace Analogue to Language (HAL), an algorithm that calculates the semantic similarity between two words by comparing the co-occurrence vectors of the two words with a Euclidean measure of distance. These approaches paved the way for the success of Distributional Semantics (DS).

Although early Distributional Semantics models display evidence of the influence of Harris's hypothesis, the distributional hypothesis is not explicitly mentioned as their theoretical foundation. Only later was the Distributional Hypothesis adopted by DS authors as a type of *a posteriori* justification for their work. Indeed, the above studies did not take into account some fundamental aspects of Harris's theories, such as the influence of syntax on the formulation of the neighborhoods of a word and the problem of non-contiguous elements of syntactic structures.

2.1 Syntax and Semantics in Harris

In Harris (1968, p. 209) there is an essential specification on the Distributional Hypothesis:

> *"...difference in meaning between words correlates with difference between them in respect to their word neighborhoods. Transformational analysis permits a more precise formulation of the neighborhoods involved: they are the arguments and the operators."*

The correlation between a word's neighborhoods and its syntactic context appears even more clearly in Harris's later works. In Harris (1988, 1991), he described language structure in terms of constraints. Each word combination is characterized by a set of constraints, "each of which precludes particular classes of combination from occurring in utterances of given language" (Harris 1991, p. 53). These constraints (*partial order*, *likelihood*, and *reduction*) act on the product of another constraint in a cascading mechanism.

The first constraint regards the *partial ordering on words* understood as "what gives a word-sequence the capacity to express fixed semantic relations among its words"

(Harris 1991, p. 5). It acts above the other constraints. It is the "essential one" (Harris 1991, p. 7) because it creates sentences.

In the partial order or Operator-Argument constraint, a word serves as the *Operator* over the other words called *Arguments*. The words of a language obtain their ability to co-occur in sentences thanks to the partial order: a word like *eat* is higher than *sheep* or *grass* because it can operate on nouns as in "*sheep eat grass*". Other operators such as *know* or *probable* are higher than eat because they can operate over it as in "*I know that sheep eat grass*". Sentences can be defined as word-sequences that satisfy this partial order. The operator-argument relations yield the meaning of the entire sentence by applying partial order relations to the meaning of the words. The sentence meaning is "the hierarchy of predicatings among the meanings of the words of the sentence" (Harris 1991, p. 8).

The likelihood constraint regards the meaning of words. For each argument word, there are some words that "are more likely than others to appear as operator on it" (Harris 1991, p. 5). In other words, the meaning of a word is determined by the selection of words (word-choice) that are operators of arguments in a given sentence (Harris 1976a, p. 263).

This constraint is strongly related to distributional analysis and the concept of *dependence*. *Dependence* is conceived as "a relation between a word and an ordered set of word classes". As exemplified by Harris (1991, p. 55), in "*the child sleeps*", the verb *sleep* depends on a word of a particular class of objects such as *Mary*, *John*, *the child*, etc. Therefore, as an argument of *sleep*, we can find a particular set of elements that corresponds to the set of Nouns. This dependency produces a *similarity* between the elements in the group. The dependence is never complete, but there are "various degrees and types of occurrence-dependence" (Harris 1954, p. 159). Among these nouns, we can find *John*, *the child*, *the dog*, and, more rarely, *the city* (*The city sleeps*), *the tree* (*Trees have to sleep each winter*). The *likelihood-gradation* between operators and arguments is a crucial relationship in language structure, and these inequalities in likelihood are not modified by transformations (Harris 1976b, p. 243).

The third constraint concerns the *reduction* of a word-sequence that helps produce more compact sentences. Certain words with a high likelihood contribute to the meaning of the sentence with a small amount of information (Harris 1991, p. 84). For example, the sequence *to come* in sentences like "*John expects Mary to come*", has a very high likelihood for the operator expects, and its reduction produces an acceptable sentence (*John expects Mary*). Harris identifies three kinds of widespread reductions. Reduction to zero (zeroing), which is the case of the example above. Reduction to affixes, as in the word *childhood*, in which the suffix *-hood* derives from the Old English *had*, "*state, condition*". Reduction to pronouns as in the sentence "*I met John, who sends regards*", which is a reduction from "*I met John; John – the preceding word has the same referent as the word before – sends regards*". "*John – the preceding word has the same referent as the word before*" is reduced to who, and, in some cases, can be zeroed ("*The money which is needed is unavailable*", "*The money needed is unavailable*") (Harris 1991, p. 81-82).

As indicated above, Harris states that each constraint acts on the product of another constraint; thus, the third constraint, reduction, acts on the product of the Likelihood constraint. The latter, in turn, acts on the product of the partial-order constraint. Hence, as affirmed by Harris, "given the meanings of the words, finding the operator- argument relations among the words of a sentence yields its meaning directly: that meaning is the hierarchy of predicatings among the meanings of the words of the sentence"(Harris 1991, p. 8). In other words, "the syntax of a sentence indicates its semantics" (Harris 1991, p. 9).

Reduction is included in the set of basic transformations (Harris 1991, p. 210). Those

basic transformations (zeroing or reduction, permutation of word-classes, single-word adjuncts, sentence nominalization, and conjoined sentences) make it possible to derive the *base sentence* (or kernel sentence) from two kinds of paraphrastic sentence: sentences with additional words (e.g. *the sheep eat grass*; *I know sheep eat grass*) and sentences with no addition but with a change (e.g. *He reads all day*; *He reads things all day*).
In all these transformations, the partial-order and the “major elements of meaning” are preserved (Harris 1991, p. 290). The word-sequence (given by the partial order) of unreduced sentences is not modified by reduction, and word-choice (resulting from likelihood and partial order constraints) is preserved under transformations. "With the preservation of word-choice comes meaning-preservation" (Harris 1991, p. 229).

These constraints suggest that, in Harris, the syntactic relation between operators and arguments yields the semantics of the sentence. Besides, the meaning of a single word depends on the likelihood that it will appear in its various operator-argument statuses. Reductions and transformations alter neither the operator-argument relation nor the likelihood inequalities.

Since a speech event is always developed in a single dimension of time, it needs a linear order that differs from the partial order. In addition to the three constraints illustrated above, Harris (1991, p. 6) hypothesizes that, after the partial order, the “words are put in one or more linear forms”.

In another paper (Harris 1968), the author affirmed that one of the relevant properties of language is the *linear order* of entities. Though operators and their operand (argument)[1] must be contiguous, Harris contemplates that “later operators on the resultant may intervene between the earlier operator and its operand, separating them” (Harris 1968, p. 16). Thus, contiguity does not refer to single words but to well-formed subsequences that constitute the sentence. The construction of the sentence, stated Harris, must be formulated on the basis of entities that are larger than words “in respect to which there are no noncontiguous phenomena” (Harris 1968, p. 32).

2.1.1 How the SD-W2 model reflects Harris’s constraints

Most DS models consider the context in its linear form when they find co-occurrences of a word. In fact, texts reflect in space the linearity of the temporal dimension in which speech is developed. However, this linear representation of a sentence does not reflect its structure, which must be described in terms of grammatical relations. By exploiting the syntactic relations emerging from a syntactic parsing process, the SD-W2 model aims to consider the three constraints mentioned above as a guideline to extract the context of words. Sentence 1 points out the differences between the two kinds of approach.

Example 1
The man who came into the bank with the gun and the mask shot the policeman.

According to Harris, Example 1 results from a set of transformation and reduction (mainly reduction to pronoun, zeroing, and conjunction) over a set of kernel sentences, each of which observes a specific partial order. The set of kernel sentences is as follows:

1. the man shot the policeman
2. someone came into the bank

[1] Harris alternates between operands and arguments

3. someone had a gun
4. someone had a mask

As indicated above, transformations and reductions do not alter the partial order, so the information yielded by the kernel sentences must be preserved in Example 1.
Classical word-window models such as HAL or COALS consider windows of 4-10 words as being context linear. They produce co-occurrence values based on the linear distance between words. In Sentence 1, for example, a five word-window selects the sequence *who came into the bank* as the context of the subject *man*. They cannot even relate the subject *man* and the operator *shot* because, in Example 1, the distance between the subject and the main operator exceeds the window size.
Unlike classical word-window models, SD-W2 reflects the original structure given by the partial order in the four kernel sentences. Considering that *someone* in 2, 3, and 4 refers to the *man* in 1, the syntactic context of the four kernel sentences in terms of syntactic distance is the same. We have distance 1 between arguments (subject and complement) and the operator and distance 2 between the subject and the complement. The model can correctly connect the argument and its operators even if they are not contiguous or if a large relative clause separates them.

Table 1
Linear and Syntactic distance between the word *man* and the other nouns of the Example 1

	came	**bank**	**gun**	**shot**	**policeman**
Linear Distance	2	5	8	12	14
Syntactic Distance	1	2	2	1	2

Table 1 shows the linear and the syntactic distances between the noun *man* and the Verbs and Nouns in the sentence. The verb shot is 12 words away from the subject and cannot be included in the context of the noun by a 5 or 10 word-window. Our model captures this relationship in the same way that it captures the relation between *man* and the verb of the relative clause, *came*.

In addition, if the sentence were subject to additional transformations (*the policeman was shot by the man who came into the bank with the gun and the mask*), the distances remain unaltered, and the context of the word *man* is preserved.
Our model takes advantage of the dependencies between the words in the sentence that emerge from the automatic parsing in order to consider non-linear relations in the context selection. In this way, we can easily relate the operator with all its arguments, even if they are non-adjacent or represented by a pronoun. Only a model with these characteristics can capture the semantic structure of the sentence because its meaning depends on both syntax and semantics and the relation between them. A distributional semantics model cannot consider the sentence as a linear concatenation of elements because the semantic structure that underlies the syntactic structures is not linear. The context of a word must be considered as its partial order and must remain unaltered after reduction or transformation.

Since the 1990s, a relative small number of dependency-based models have been presented, (Padó and Lapata 2007; Grefenstette 1992; Lin 1997; Strzalkowski 1994). These models seek to exploit syntactic dependency so as to obtain a more precise simulation of Human knowledge-learning phenomena. There is no empirical evidence for the supremacy of this kind of model in general tasks (Kiela and Clark (2014), and Lapesa and Evert (2017) reports substantially comparable results). In addition, syntactic models generally require a large amount of pre-processing. Nevertheless, thanks to improvements in syntactic parsers and computing power, we feel that using syntactic data to perform similarity computation is of primary importance.

In the next section, we will provide a rapid overview of the DS models that have most influenced our work.

3. Related Works

In section 2, we analyzed Harris's theories on meaning and the relation between syntax and semantics and how he directly or indirectly influences later theories.

Harris's distributional hypothesis is rooted in structuralist theories and in Saussure's concept of *valeur* (Sahlgren 2008, p. 5). The differential view of meaning that characterized Harris and, earlier, Bloomfield is based on the idea that signs are identified by their functional differences (the *sign's valeur*). A sign assumes a *valeur* by virtue of its "being different from other signs"; it therefore emerges only in a system and cannot exist in isolation. Saussure considered two kinds of relation in which functional differences emerge. *Syntagmatic* relations concern connections between words that co-occur (*in praesentia*); *paradigmatic* relations concern substitution, and related words that do not co-occur (*in absentia*).

According to this difference, Sahlgren (2008) classified distributional models as Syntagmatic or Paradigmatic models.

The first family of models focuses on Sentence Meaning. These models study polysemy, disambiguation, and semantic compositionality from a distributional point of view. Disambiguating polysemous words cannot be addressed with a traditional approach based on formal semantics, such as the standard Distributional Semantics Models (Baroni, Bernardi, and Zamparelli 2014). There are two predominant approaches: the first encodes all relevant information for a given word and then uses context to find the right meaning. The second builds different vectors for each word sense (Boleda 2020).

Related to the concept of paradigmatic and syntagmatic relations is the classification of first-, second- and third-order techniques produced by Grefenstette (1994). The author defines first-order techniques as those that look at the local context to discover what other words can be found among the neighbors of a given word. Second-order techniques look for terms that share the same environments. Third-order techniques create semantic groups of similar words by manipulating the list of similar words produced by a second-order technique.

Distributional Semantics Algorithms based on Harris's distributional hypothesis can be classified in the second family of models, paradigmatic models, or second and third-order techniques.

As pointed out in section 1, these models can be classified by using different criteria (Lenci 2018): if we consider the context selection, we can classify them into Word-Based models and Document-Based models. While document-based models consider a whole document as the context, word-based models take a variable number of words.

In the last few years, several models based on neural network algorithms have appeared. Since the introduction of Word2Vec (Mikolov et al. 2013b), these so called *predict models* (Baroni, Dinu, and Kruszewski 2014), have demonstrated their superiority over traditional models.

More recently, deep neural networks have been applied to traditional and predict models in order to overcome the idea that each token must correspond to a vector

(Peters et al. 2018): these latest-generation models represent a word with a number of vectors equivalent to the different sentence contexts in which it appears. For this reason, these models are called *contextualized word embeddings*.
Contextualized models work by learning the vectors as a function of internal states of a pre-trained encoder (Chersoni et al. 2021) such as Long Short Term Memory (LSTM) for feature-based approaches (Peters et al. 2018), or Transformers for fine-tuning approaches (Devlin et al. 2019). In particular, BERT (Devlin et al. 2019) and ELMo (Peters et al. 2018), became very popular in the last years because offers generalized solution to many computational linguistic tasks with very high performances.

The model proposed in this paper does not take this kind of technology into consideration. We aim to demonstrate that the influence of syntax on the generation of semantic word matrices could improve the results of DS models, regardless of the family the model belongs to.

As was illustrated in section 2, a large part of models consider words that belong to the same document or sentence as co-occurring. These models do not make use of linguistic data. However, many other models are built in such a way that linguistic knowledge affects the collection of distributional information. These models aim to use part of speech tags, lemmas, or dependencies. Since the proposed model is a word-based dependency model that explores paradigmatic relations, we will present a rapid overview of Distributional Semantics algorithms that influence our work.

3.1 Window-Based Models

Our overview begins with Hyperspace Analogue to Language (HAL) (Lund and Burgess 1996), which is considered one of the most influential Distributional models (Lenci 2008).

In HAL, the semantic similarity between two words is calculated by comparing word-vectors with Euclidean distance measures, extracted from a large co-occurrence matrix. HAL reads the corpus through an n-words window to generate the co-occurrence matrix. The window size suggested by the authors ranges between 5 and 10 words, and the corpus must include a large set of heterogeneous texts.

The authors use a lexicon of the 70,000 most frequently used terms of English to generate a HAL matrix with a dimension of 70,000 X 70,000 (Burgess 1998). Each word vector is processed with a multidimensional scaling algorithm to transform it into a bi-dimensional pictorial representation of the word. This procedure generates semantic knowledge by grouping semantic neighbors and grammatical knowledge. The corpus used to generate the matrix is 300 million words of English text from Usenet newsgroups. This methodology makes it possible to represent the semantic meaning of words and bring out the characterization of a variety of aspects of lexical ambiguity (Burgess 2001). HAL exerted a major influence on many later models (Audet and Burgess 1999; Azzopardi, Girolami, and Crowe 2005; Rohde, Gonnerman, and Plaut 2006).

In particular, *Correlated Occurrence Analogue to Lexical Semantics* (COALS) (Rohde, Gonnerman, and Plaut 2006) achieves considerably better performance levels. In HAL, the authors believe that high-frequency columns make an excessive contribution to the distance measure. COALS employs a normalization strategy that solves this issue. The model is set on a flat 4-word window and computed on the 100,000 most frequent words as columns and 1 million rows. Once the 4-word window completes the matrix building process, the co-occurrence value is replaced with a value calculated as a Pearson Correlation between each row. The Pearson Correlation measures the linear dependence between two variables. It is one of the first measures of correlation and remains one of the most widely used measures of relationship (Schober, Boer, and Schwarte 2018). The Pearson Correlation generates values in a -1 to 1 range, in which -1 is a total negative correlation, 1 is a total positive correlation, and 0 represents the complete absence of correlation. The authors transform all negative values into 0 and square all other values. By setting all negative values to 0, the authors obtain a scattered matrix, losing information on anti-correlated words that do not generate similarity values between words. Conversely, by squaring all positive values, the importance of many small values is exalted in comparison to the few larger ones.

As regards vector length, the authors choose to eliminate purely syntactic words such as determiners or punctuation symbols, using a 14,000 columns matrix. Finally, vector similarity is calculated by using the Pearson Correlation once again.

The model was tested on several tasks, including word-pair similarity ratings, multiple-choice vocabulary tests, yielding a better performance than other state-of-the-art models. The results were also confirmed in Jurgens and Stevens (2010), who compare different algorithms.

A different Window-Based family of models employs a Random Indexing approach (Kanerva, Kristoferson, and Holst 2000). Random Indexing produces low-dimensional random vector representations of each context. When the word-window scans the corpus, each time a word occurs in a context, the random vector is added to the context vector (Sahlgren 2005). Since the dimensionality of the random vector is reduced, the context vectors will also have the same dimension. This method makes it possible to build the matrix incrementally, with low-dimension and with any kind of context selection method.

Lapesa and Evert (2014) investigated the impact of various Word-window model parameters on a number of traditional semantic tasks. Three parameters appear to have a particularly significant impact on a model's performance: *score* (how the algorithm assigns a co-occurrence value to the words in the word-window), *transformation* (how the co-occurrence scores are then transformed so as to reduce the features' asymmetry) and *distance metric*.

The impact of those parameters, and in particular of *transformation* can explain the better performance of COALS compared to HAL: since the other parameters are similar for both models, the introduction of a matrix transformation is the primary distinction between them. While HAL does not provide any kind of transformation of the matrix, COALS employs Pearson's transformation.

Other parameters (*corpus*, *window size*, *dimensionality reduction*) also exerted an influence, but they varied more widely in response to the task. For example, the *Difference of Means* between reduced and unreduced models is quite substantial for the TOEFL task; for the other tasks, the use of the WaCkypedia corpus (Baroni et al. 2009) yields better results.

3.2 Dependency-Based Models

Dependency-Based Distributional Semantics, also known as syntax-based distributional semantics, inspires a class of algorithms that use linguistic annotation to improve the results of similarity measure extraction. In general, we can consider these models as belonging to word-based models because only words belonging to the same sentence are included in the context. Unlike HAL, this kind of method does not assign co-occurrence values according to nearness between words, but they take advantage of the syntactic relations shown by a syntactic parser.

Regardless of the amount of pre-processing required, the differences between syntactic models and word-window models in terms of performance are difficult to judge. Traditional Word-window models, also known as bag-of-words models, generally achieve the best performance in classification tasks, while *bag-of-arguments* models (Dependency models) perform better in predicting argument expectations (Chersoni et al. 2017). Levy and Goldberg (2014) train the word-window model *SkipGram* (Mikolov et al. 2013b) and perform their experiments with a dependency-based context. They show that the dependency-based context yields a different embedding, such as *functional similarities of a cohyponym nature*.

The first dependency-based algorithm to return promising results in distributional semantics was presented by Grefenstette (1992). The paper's idea was to take advantage of the growing availability of syntactic parsers to select the syntactic context of words. The model, called *Sextant*, derives similarity measures that consider the overlapping of all contexts associated with a target word over the corpus.

Other influential syntactic models were presented by Strzalkowski (1994) and Lin (1997): Strzalkowski (1994) presents a dependency-based methodology included within an information retrieval task. The authors propose the extraction of a set of head+modifier pairs from a parsed text, which are used as occurrence contexts for each term included in them. Two terms that share some modifiers but appear in a few distinct contexts receive a similarity coefficient of between 0 and 1. Lin (1997) proposes a Word Sense Disambiguation algorithm based on a Similarity Measure calculated through a syntactic context. The local context of a word is defined as a triple of dependency relations in which the word is the head or the modifier. The authors construct Local Context Databases by extracting this kind of relation and using word frequency and the likelihood ratio to give a distance value. Each target word is described as a triple (type, word, position) and a set of word-frequency-likelihood-triples.

Inspired by the works of Lin and Strzalkowski, Padó and Lapata (2007) developed a model based on the notion of *paths*. Paths are sequences of dependency edges that connect two words, the use of which makes it possible to represent both direct and indirect relationships between words. There are three new parameters related to paths: the *Context selection function* determines which path in the dependency graph contributes to the representation of the target word; the *path value function* assigns weights to paths, for instances, giving more weight to paths containing subjects and objects; the *basis mapping function* establishes the size of the semantic space. In their work, the authors list three different context selection functions, minimum, medium, and maximum, respectively of length 1, length <=3 and length <=4, and three path value functions: plain, which assigns 1 to every path, length, which assigns a value inversely proportional to the length of the path and gram-rel, which ranks paths by using a value that reflects the salience of their grammatical relations (i.e., subjects are more salient than objects). The authors also define an *optimal dependency-based model* which uses the medium context selection function and the length path value function, with 2000 basis elements. They train the model on the *British National Corpus* (100 million words) and test it on three tasks: Single-word Priming, Detection of Synonymy, and Sense Ranking. The model achieves performance levels comparable to or higher than state-of-the-art models in all the selected tasks.

More recently, Baroni and Lenci (2010) proposed an approach called *Distributional Memory* in which the authors seek to solve the problem of building a different distributional model for each different semantic task. The methodology adopted entails the extraction of co-occurrence as a ternary geometrical object of the kind word-link-word, called the third-order tensor. The tuple word-link-word is made up of two content words and a syntagmatic co-occurrence link between them: for example, the tuple <*marine,use,bomb*> denotes that the word *marine* co-occurs with the word *bomb*, with the word *use* representing the syntagmatic link between the two.

Distributional Memory provide two different models: the *dependency model* uses a set of links for *noun-verb*, *noun-noun* and *adjective-noun* pairs, which includes *verbs* (*the soldier is reading a book* → <*soldier,* verb, *book*>), the *subject of intransitive verbs* (*the teacher is singing* → <*teacher,* sbj_intr, *sing*>), the *noun modifier* (*good teacher* → <*good,* nmod, *teacher*>), etc.

The *lexical model* includes complex links, which take into account the morphological features of the pair words: *POS*, *number*, *tense*, *presence of articles*, *adjectives*, *adverbial modifier*, *auxiliary* or *modal verbs*. For example, the sentence *The tall soldier has already shot* is represented by the tuple <*soldier,* sbj_intr+n-the-j+vn-aux-already, *shot*>. The suffix of the link shows that the first word (*soldier*) is a singular noun (*n*), definite (*the*) and has an adjective (*j*), and that the second word (*shot*) is a past-participle (*vn*) with an auxiliary (*aux*) and is modified by an adverb (*already*).

Subsequently, matrices are generated directly from the tensor to perform a specific semantic task in a defined space. The model was tested on different semantic tasks and achieved a performance that, in some cases, was slightly lower than other models constructed ad hoc for the task. Nevertheless, the advantage of using a single general model that does not need to be retrained for each new task compensates for the lower performance.

Dependency-based models have also been tested on a variety of tasks to understand how different parameters affect their performance Lapesa and Evert (2017). The Dependency-based models work similarly to the window-based models in terms of performance and best values for a significant number of parameters (metric, score, transformation).

4. The SD-W2 algorithm

In order to perform a distributional analysis and calculate the similarity values from the context of the words, we choose to include a level of syntactic analysis in our model. This makes it possible to draw the real connections between words and overcome the linear vision of the sentence adopted by the word-window models.

These models extract similarity among words by calculating the similarity of their likelihood: if two words appear near the same group of words (i.e. they have similar contexts) in large corpora, then they have similar meanings. Word-based models calculate the context as a connection value between a word and all the words immediately adjacent to the target word or within a certain distance from it.

Nevertheless, as highlighted in section 2, Harris explicitly states that analysis of the meaning must rely on the first constraint (partial order). The partial order constraint acts over different hierarchies of linguistic elements: at the higher level, it works on operators that act over lower operators (i.e the verb *said*, which acts over other operators such as *eat* in sentences like "*I said that sheep eat grass*"); it acts on operator-argument relations (i.e. the verb *sleep* and its argument *child* in "*the child sleeps*"); but there also exists a hierarchical relation between the noun *reading* and the noun *book* in a sequence like *the reading of the book*. Harris (1957), assumed that the sentence "*the reading of the book is fast*" results from a set of transformations over two kernel sentences:

- k_1: the reading is fast

- k_2: someone read the book

The transformation involved are the following:

- $S \leftrightarrow N$ of k_2: the reading of the book by someone
- k_1 overlap with k_2: the reading of the book by someone is fast
- zeroing of "by someone": the reading of the book is fast

In other words, the k_2 kernel is nominalized (from *to read* to *the reading of*) and is overlapped with k_1. Finally, reduction allows the sequence *by someone* to be deleted because it brings a very small amount of information (there is always someone that reads a book).

Besides, reduction and transformations hide elements that appear with high frequency values in specific contexts and change the shape of a sentence, leaving syntactic relations unaltered. In this way, *reading* and *book* were also involved in an operator-argument relation and must be taken into account in the semantic analysis of the sentence. At a syntactic level, the distance between *reading* and *book* in the final sentence corresponds to the distance between *read* and *book* in k_2. It is the nature of the relationship that has changed.

Based on these assumptions, the proposed model attempts to extract co-occurrence values by considering the syntactic connections between words, regardless of typology or direction. The underlying idea is that the *syntactic context* of a word can be calculated on a parsed text by considering a measure derived from the concept of Syntactic Distance (Liu, Xu, and Liang 2017). As a quantified value, it works as a word-window that scrolls the text, not in its linear order but in its syntactic *partial order*. In exactly the same way as other models, the syntactic distance is converted into a numerical value which propagates through the network of relations described by the parsed text as shown in figure 1.

As illustrated by the figure, the distance is equal to the number of nodes in the syntactic sentence graph separating the target word from the other words in the sentence. At each distance, there may appear as many words as there are incoming and outgoing connections for a node.

4.1 Description of the Algorithm

The algorithm relies on the input of three external elements:

1. a base-dictionary that includes all the terms for which a vector representation is sought. We used a non-flexed dictionary and each vector will represent a single Lemma;
2. a dimension-dictionary that includes the terms representing the dimensions of each vector, i.e the columns of the matrix. This dictionary must also contain non-flexed terms;
3. a collection of documents in CoNLL format. CoNLL (Buchholz and Marsi 2006) provides a great deal of linguistic information about the text in table form. The rows of CoNLL tables represent the words that make up the document. The columns include an ID number, the FORM or token, LEMMA, universal POS Tags, HEAD, which indicates the ID of the

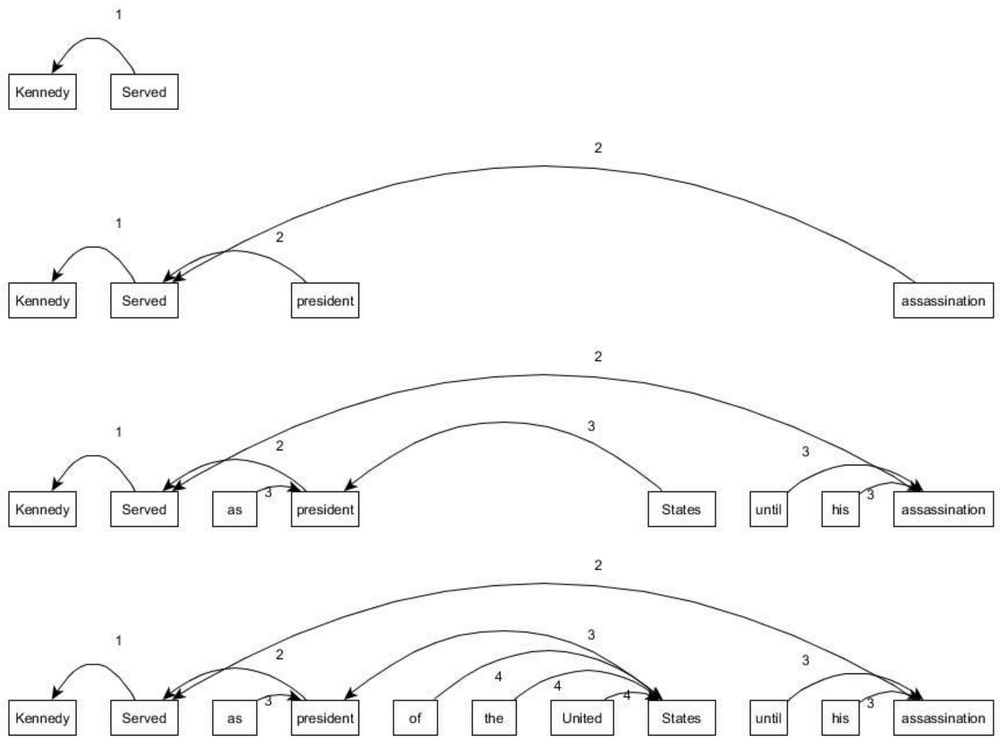


Figure 1
Syntactic Distance values for the word *Kennedy* in the sentence "Kennedy served as president of the United States until his assassination"

headword, and DEPREL, which indicates the nature of the dependency relation.

The algorithm proceeds by mapping the two dictionaries with a number that corresponds to the column/row of the matrix. Then, the algorithm takes into consideration a single sentence.

Algorithm 1 reports a description of the method that converts the input CoNLL sentence into a Sentence Graph Structure (SGS).

Algorithm 1 generation of the SGS from the CoNLL Sentence

```
Input: conllSentence
for line in conllSentence do
    add [line(ID),line(HEAD)] to SGS
    add [line(HEAD),line(ID)] to SGS
end for
Output: SGS
```

SGS is an edge graph in which the connections are represented by two values: the ID (source) and the HEAD (target) of each CoNLL row. In the SGS, connections have no direction because the syntactic distance is calculated for each pair of connected elements, regardless of the nature and direction of that connection. In order to represent the bi-directionality of the SGS simply, the algorithm also inserts the inverse connection of each arc into the graph structure.

By way of an example, we consider the sentence in CoNLL format of table 2.

Table 2
Parser data in CoNNL format of the sentence "Kennedy served as president of the United States until his assassination"

ID	FORM	LEMMA	POS	HEAD	DEPREL
1	Kennedy	Kennedy	NNP	2	nsubj
2	served	serve	VBD	0	ROOT
3	as	as	IN	4	case
4	president	president	NN	2	obl
5	of	of	IN	8	case
6	the	the	DT	8	det
7	United	united	NNP	8	compound
8	States	states	NNP	4	nmod
9	until	until	IN	11	case
10	his	his	PRP$	11	nmod:poss
11	assassination	assassination	NN	2	obl

The word *president* points to the word *Kennedy* and, consequently, they are considered to have distance 1; but *president* also has distance 1 with the words that point to it (*as*, *state*). From the word *president*, whose ID number is 4, generates the arc {4,2}; since the connections are bidirectional, it also generates {2,4}. In addition, the words that point to *president* generate the arcs {3,4}, {4,3}, {8,4}, and {4,8}.

The SGS of the sentence in tab 4.1 will includes the following list of edges:

{0,2};{1,2};{2,0};{2,1};{2,4};{2,11};{3,4};{4,2};{4,3};{5,8};{6,8};
{7,8};{8,4};{8,5};{8,6};{8,7};{9,11};{10,11};{11,2};{11,9};{11,10}

Once the algorithm has processed the dictionaries and created the SGS, it removes all the SGS edges that involve the ROOT (all the pairs that include a zero). At this point, the algorithm starts the syntactic context analysis by inserting the sentence co-occurrence values into the matrix. Algorithm 2 describes the syntactic co-occurrence analysis.

The loop takes as input the SGS, the target word and a structure that maps the ID of each word with its POS and Lemma. It also needs two parameters:

- **Syntactic Distance**: the variable *windowSize* corresponds to the size of the syntactic window taken into account. This value ranges from 1 to 5.
- **Weighting function**: the function that determines the weight to assign to co-occurring words according to their distance.

In the first part of algorithm 2, it extracts the words directly connected with the target word. It assigns the value 1 to the connected words and stores their ID values

Algorithm 2 SyntCoOccAnalysis (SGS(Sentence), targetWord(ID,Lemma,PoS))

```
Input: SGS[value.0,value.1], targetWord[ID,LEMMA,POS]
Parameters: windowSize, weightingFunction
linearDistance = 1
targetWordMass = setMass(targetWord[POS])
for edge in SGS do
   if edge[value.0] is equal to targetWord[ID] then
      distances[edge[value.1]] += linearDistance
      add edge[value.1] to propagation
      remove targetWord[ID] from propagation
      windowSize = windowSize-1
   end if
end for
while windowSize > 0 do
   linearDistance =+ 1
   for id in propagation do
      for edge in SGS do
         if edge[value.0] is equal to id then
            distances[edge[value.1]] += linearDistance
            add edge[value.1] to propagation
            remove id from propagation
            windowSize = windowSize-1
         end if
      end for
   end for
end while
coOccurrenceValues = WeightingCoOcc(distances,weightingFunction)
Output: co-Occurrence Values of a Sentence (coOccurreceValues)
```

to continue the propagation. Then, according to the value of *windowSize*, it starts a new loop for all the IDs in the propagation list.

The algorithm assigns co-occurrence values after calculating the distances between the target word and the other words in the sentence (Algorithm 3). The Weighting Function may be of two types: a linear function that assigns a decreasing value to the words as the distance increases the target word or a GRAV[2] function based on the POS of the target word.

Finally, the algorithm actualizes the general matrix, adding the values generated for the target word for each context word and repeating the loop for the next target word.

When the corpus has been entirely processed, the algorithm converts the co-occurrence matrix into a correlation matrix. In the COALS algorithm, the Pearson Correlation is performed over the original matrix so as to generate the conditional rate instead of the raw rate of word-pair co-occurrence. The authors claim that computing *Pearson's correlation* between the occurrence of a word *a* and a word *b* can express the tendency of *b* to occur "more or less often in the vicinity of a than it does in general".

[2] We will illustrate the GRAV function in section 4.2.2

Algorithm 3 WeightingCoOcc (SyntDistances(Sentence), weightingFunction)

```
Input: distances Map
Parameters: weightingFunction
for key in distances do
    contextWord = sentenceWord[key].value(LEMMA)
    if weightingFunction is Linear then
        coOccurrenceValues[contextWord] = (distances[key]*-1)+(WindowSize+1)
    else if weightingFunction is GRAV then
        coOccurrenceValues[contextWord] = $Mass^2/distances[key]$
    end if
    if coOccurrenceValues[contextWord]<0 then
        coOccurrenceValues[contextWord] = 0
    end if
end for
```

This normalization converts the co-occurrence values into values that range between -1 and 1. Converting all resulting negative correlations, which represent anti-correlated words, to 0, the matrix becomes more sparse and the model's performance may improve. Rohde, Gonnerman, and Plaut (2006) compared the COALS algorithm with a similar algorithm like HAL, which differs from the former mainly in this feature, obtaining considerably better results.

Taking into account the sentence presented in table 2, a word window of 5 and a Linear weighting function, we obtained the matrix shown in table 3.

Table 3
The Matrix generated by the presented model of sentence in table 2

	Kennedy	served	as	president	of	the	United	States	until	his	assassination
Kennedy	0	5	3	4	2	2	2	3	3	3	4
served	5	0	4	5	3	3	3	4	4	4	5
as	3	4	0	5	3	3	3	4	2	2	3
president	4	5	5	0	4	4	4	5	3	3	4
of	2	3	3	4	0	4	4	5	1	1	2
the	2	3	3	4	4	0	4	5	1	1	2
United	2	3	3	4	4	4	0	5	1	1	2
States	3	4	4	5	5	5	5	0	2	2	3
until	3	4	2	3	1	1	1	2	0	4	5
his	3	4	2	3	1	1	1	2	4	0	5
assassination	4	5	3	4	2	2	2	3	5	5	0

The matrix shown in table 3 is dense. Matrix density is particularly pertinent to short sentences, but the algorithm generally produces denser matrices with high values of word-window because, while words in syntactic structures are much more interconnected, a value higher than 5 tends to propagate throughout the sentence. This is an obvious consequence of using syntactic parsing data in matrix construction (Sahlgren 2008).

Table 4 shows the results of applying the Pearson Correlation to the Matrix presented in Table 3. Matrix density decreases markedly in Table 4. The matrix becomes even more sparse when lower values of word-window are used.

Table 4
The matrix after Pearson Correlation

	Kennedy	served	as	president	of	the	United	States	until	his	assassination
Kennedy	0	0,219	0,086	0,116	0	0	0	0	0,168	0,168	0,179
serve	0,219	0	0,112	0,105	0	0	0	0	0,192	0,192	0,179
as	0,086	0,112	0	0,201	0,119	0,119	0,119	0,136	0	0	0
president	0,116	0,105	0,201	0	0,146	0,146	0,146	0,133	0,014	0,014	0
of	0	0	0,119	0,146	0	0,248	0,248	0,252	0	0	0
the	0	0	0,119	0,146	0,248	0	0,248	0,252	0	0	0
united	0	0	0,119	0,146	0,248	0,248	0	0,252	0	0	0
states	0	0	0,136	0,133	0,252	0,252	0,252	0	0	0	0
until	0,168	0,192	0	0,014	0	0	0	0	0	0,295	0,298
his	0,168	0,192	0	0,014	0	0	0	0	0,295	0	0,298
assassination	0,179	0,179	0	0	0	0	0	0	0,298	0,298	0

4.2 SD-W2 parameter selection

In a preliminary experimentation phase, we tested different criteria for the parameter selection of the presented algorithm. These parameters are:

- Syntactic Distance
- Weighting function

In addition, we tested the Singular Value Decomposition (SVD) algorithm (Rohde 2002) in order to vary the dimensionality of the final matrix. SVD is a method for the linear decomposition of a matrix into independent components adopted for the first time by Landauer and Dumais (1997) in Distributional Semantics for Latent Semantic Analysis. The LSA model uses the SVD algorithm to produce a better simulation of human word-learning. The authors claim that SVD *embodies the kind of inductive mechanisms that they want to explore and provides a convenient way to vary dimensionality*. Since SVD did not greatly change performance in our preliminary test, we decided not to add a dimension reduction algorithm to our model.

4.2.1 Syntactic Distance

The selection of window size in word-based distributional semantics models can consider a neighborhood ranging from one to 1000 words (Sahlgren 2008). Schutze (1992a) proposes a window size of 1000-1200 words, claiming that word size is more important than the number of words taken into account in context construction. Yarowsky (1992) and Gale, Church, and Yarowsky (1995) use 100-word windows. Lund and Burgess (1996) use 10-word windows in HAL and Rohde, Gonnerman, and Plaut (2006) in COALS, suggest using 4-word windows. Although there are no word-windows in Syntactic methods, they extract co-occurring words from a dependency graph by defining a list of paths. The length of this paths plays the same role as the dimension of the word-window in linear models.

With this work we present a syntactic model in which we replace specific dependency paths with a generic syntactic window in which all the words related with a target word within a variable syntactic distance are included in its context. The value of Syntactic Distance, in this way, work exactly as a variable word-window, with the difference that it was unclear how many words the model would include in the context.

For example, if we can find more than one word in a sentence at a distance of 1, the number of words taken into account grows when this distance value increases.

The distance value used in our experiment ranges from 1 to 5.

4.2.2 Weighting Functions

We tested the system using a *linear weighting function* in which the co-occurrence value ranges from the dimension of the word-window to zero, decreasing once the syntactic distance grows. With $d = SyntacticDistance$ and $w = window - size$ the co-occurrence value c is calculated as:

$$c = (-d + (w + 1))$$

In the sentence *Kennedy served as a president of the United States until his assassination*, taking into account the word *Kennedy* as Target Word and a window-size of 2, the algorithm assign $-1 + (2 + 1) = 2$ to syntactically adjacent words (*served*), $-2 + (2 + 1) = 1$ to words at distance 2 (*president* and *assassination*), $-3 + (2 + 1) = 0$ to word at distance 3, and so on, setting all the negative values to zero.

In order to improve the variability of co-occurrence values, we also tested a different function, related to the words' syntactic features and using the parser graph. We were inspired by the idea that some words with certain POS tags (i.e. function words) tend to be very frequent and do not convey semantic information (Rohde, Gonnerman, and Plaut 2006). In COALS, these words were excluded from the final matrix. Our aim is to preserve this information but introduce proportional weights for each POS.

The parsed sentences are graphs in which words are nodes and relations are directed edges. By considering POS tags as nodes, we extract the total of the relationships in which each POS tag is involved in a section of one million words of the British National Corpus (BNC), parsed with the Stanford Core-NLP Parser Package.

Since we convert dependency graphs into undirected graphs (we take into account relations both pointing towards a node and starting from the node), we choose to use the total percentage of relations (in+out) as the *Mass* of a word. In our opinion, this value reflects the centrality of the POS tag in the sum of sentence networks of the corpus and proposes a set of values with greater significance and variability.

The main idea is to give each word a weight based on its influence on the syntactic graphs. Nouns and Verbs, for example, have a high *Mass* value that reflects their centrality in the structure of the sentences.

Definition 2
The **Mass** of a word is equivalent to the ratio of the number of incoming and outcoming arcs of a given POS and the total number of relations in a 1 million word Corpus extracted from the BNC.

For example, Nouns are involved in 41% of the relations in the first one million words of the BNC. This means that out of 100 arcs in the sum of the dependency graphs, 55 point to and 27 start from a Noun. If we observe the dependency graphs, we will see that Nouns are pointed to by Determiners (*the book*), Adjectives (*beautiful girl*) and other Nouns (*city center*). Conversely, they point mainly to Verbs, Nouns and Prepositions. If we take Determiners or Adjectives into account, these are involved in 5% and 7% of edges and, in the vast majority of cases, they point only to Nouns.

When we score the co-occurrence of the terms included in our matrix, we give higher values to categories that we consider central to our semantic analysis, without completely eliminating categories that include non-content words. In the final matrix, this difference only affects rows, because the score is influenced only by the mass of the target word. In addition, we square the values so as to increase the difference between the POS tags and to obtain better results.

The influence of the *Mass* of a word must decrease as the distance increases, so the weight function, called GRAV, is calculated using the following formula:

$$GRAV = Mass_t^2 / Distance_{t,w}$$

$Mass_t$ indicates the weight of the POS tag of the Target Word and $Distance_{t,w}$ is the syntactic distance between the target word and the co-occurring word. In this perspective, each word may be considered an object with a certain *syntactic Mass* and produces an *attraction* over its neighbor words that is stronger if the POS of the word tends to be central in sentence networks. The attraction decreases as the distance increases.

In the sentence *Kennedy served as a president of the United States until his assassination*, taking into account the word *Kennedy* as Target Word and a window-size of 3, the algorithm assign $41,26^2/1 = 1.702,3876$ to syntactically adjacent words (*served*), $41,26^2/2 = 851,1938$ to words at distance 2 (*president* and *assassination*), $41,26^2/3 = 567,4625$ to word at distance 3. Conversely, the word *the* will obtain a co-occurrence value of $5,32^2/1 = 28,5156$ with its adjacent word *United*, $5,32^2/2 = 14,2578$ with words at distance 2 and $5,32^2/3 = 9,5052$ with words at distance 3.

4.3 Best Configuration

The algorithm presented in the previous section was developed in Java, using the *sspace* package developed at the Natural Language Processing group at UCLA[3]. The package contains algorithms and tools for constructing a distributional model and a set of compiled well-known classic algorithms such as LSA, HAL, DVS, and COALS.

In order to test the parameter of the model, we use the British National Corpus (Leech 1992), a 100 million-word Corpus of English, including written and spoken language. The corpus was parsed with the Stanford Core-NLP Parser Package (Manning et al. 2014).

The dictionary we used as Base-Map includes more than 18,000 words with high-frequency values extracted from the BNC[4] (more than 400 occurrences in BNC), which correspond to 12,024 lemmas.

With a view to testing our model, we defined an *optimal model* with a parameter setting that maximizes the experimental results. To test the parameter selection, we used the Rubenstein and Goodenough similarity test (Rubenstein and Goodenough 1965), as suggested by Padó and Lapata (2007). The original test calculated the correlation between the evaluations of semantic similarity performed by groups of humans on two lists of 24 theme words. The experiment involved 65 noun pairs scored on a 0-4 scale. The original model calculated a Pearson correlation (Pearson's *r*) coefficient of 0.85 when applied to similarity ratings between annotators.

We obtained the best results with no matrix reduction applied. The differences between weighting functions and syntactic distance (D) are shown in table 5.

[3] The sspace package is freely downloadable at https://github.com/fozziethebeat/S-Space/wiki

[4] Frequency list download at http://www.kilgarriff.co.uk/bnc-readme.html

Table 5
Evaluation of different parameters application on Rubenstein and Goodenough test

Syntactic distance	Linear WF	GRAV WF
1	0.65	0.64
2	0.656	**0.661**
3	0.63	0.64
4	0.61	0.63
5	0.59	0.62

The results presented in table 5 show a minimal variation between the application of the two weighting functions, with a slight advantage for the GRAV function. Conversely, the syntactic distance shows bigger variations with a clear propensity for models with the syntactic distance set as 2. The selected parameters were:

- words and Dimensions: 12,024
- Distance: 2
- Weighting function: GRAV

Once the parameters producing the best results are established, we also train the model on a larger corpus, the *WaCkypedia English corpus* (Baroni et al. 2009), a 2009 dump of English Wikipedia, cleaned and parsed with MaltParser (Nivre, Hall, and Nilsson 2006), of about 800 million tokens.

5. Experiment

This section presents a series of experiments on which the methodology described in section 4 was tested. As announced in section 1, our results on three tasks will be compared with other word-window models. Since we found an optimal configuration for our parameter, we retrain the model using a larger corpus.

The experiments we report in the paper are related to the classic semantic tasks addressed by many authors in DS literature:

- Semantic Similarity: a set of experiments in which the algorithm must express a similarity value between two words in a list of pairs already classified by humans. The correlation between the values given by the model and the human's values represents the algorithm's assessment score.
- Synonymy: this kind of text is based on synonymy tests generally proposed to foreign students of English during their assessment. The test consists of choosing the correct synonym for a word from four alternatives.
- Single-Word Priming: the test consists of finding the strongest association between a set of words representing six different lexical relations (synonymy, antonymy, super-subordination, category coordination, conceptual association, and phrasal association).

- In addition to these experiments, we will introduce a new task related to the concept of *selection* as conceived by Harris. This measures the similarity of a group of nouns belonging to a specific class with a verb that selects that class as the subject or the object.

In order to gain a clearer idea of the obtained results, we compared the two trained models (WaCkypedia and BNC) with other state-of-the-art models:

- Contextualized Models such as BERT (Devlin et al. 2019) or ELMo (Peters et al. 2018), as reported in Lenci et al. (2022) and Wang, Cui, and Zhang (2021);
- the results of similar models such as COALS and DVS, as reported in its original papers and by Jurgens and Stevens (2010);
- the results of classic models such as LSA and HAL, as reported by various sources;
- the results of COALS and Word2Vec (Mikolov et al. 2013b, 2013a) trained on the BNC corpus.

The data set and the experiment on the *argument selection task* will be presented in section 5.4; section 5.1 shows the results of our model on Semantic Similarity Task, in 5.2 we present the experiments on synonymy tasks and in 5.3 we replicate the semantic priming experiment presented in Padó and Lapata (2007) using our model.

5.1 Semantic Similarity Task

Semantic Relatedness is an important research topic in NLP (Taieb, Zesch, and Aouicha 2020). To verify the effectiveness of semantic relatedness extraction methods, the computational results are usually compared with human judgments. The cost of manual annotation of relatedness values limits the size of this kind of evaluation data set. Besides, a careful selection of the words is required.

We decided to test our algorithm on four Semantic Similarity data sets that have been used as a test set by many other authors. In particular, we tested our optimal model on the following data sets:

- **Rubenstein and Goodenough similarity pairs** (Rubenstein and Goodenough 1965) (RG65): this data set, described in section 4.3, is one of the most frequently used in evaluating DS models on semantic similarity. We compared our results with the results reported by Padó and Lapata (2007); Rohde, Gonnerman, and Plaut (2006); Landauer and Dumais (1997); Lund and Burgess (1996) and compared the evaluation of the same models trained on different corpora presented by Jurgens and Stevens (2010). In accordance with Rohde, Gonnerman, and Plaut (2006), we also tested the model on a reduced RG data set of 52 pairs of words, produced by deleting 5 ambiguous words.
- **Miller and Charles ratings** (Miller and Charles 1991) (MC30): this is another common similarity data set, which includes 30-word pairs of the RG65 manually evaluated by 38 subjects. The words selected for the MC30 data set have higher frequencies than the original RG set. For this subset,

we used both the original one and a reduced version with 5 ambiguous words deleted and 24 pairs.

- **WordSimilarity-353 Test Collection** (Finkelstein et al. 2001) (WS353): this data set includes 353 pairs rated by 13 or 16 subjects on a 0-10 scale. The set includes the MC30 pairs, proper names (such as *Arafat* or *Maradona*), word associates that are not synonymous (*tennis-racket*), adjectives, or gerunds. The words of WS353 are, in general, more common than those in RG.
- **SimLex999** (Hill, Reichart, and Korhonen 2015) (SL999): SimLex-999 is a gold standard resource for semantic similarity tasks. Five hundred native English speakers produced the resource: it contains 999 adjective, verb, and noun concept pairs. The experiment was designed as shown in Hill, Reichart, and Korhonen (2015), in order to compare the optimal model with the performance presented in that paper on the whole set and *abstract-concrete* subset and *Adjective-Noun-Verb* subset.

Table 6
Comparison of different algorithms on different Semantic Similarity Data Sets

Algorithm	Corpus	RG65	MC30	WS353	SimLex999
SD-W2	BNC	0.682	0.605	0.527	0.303
COALS	BNC	0.569	0.453	0.427	0.22
DVS	BNC	0.62	-	-	-
W2V (CBOW)	BNC	0.678	0.647	0.566	0.324
SD-W2	Wikipedia	**0.842**	**0.76**	0.614	0.394
BERT.L4	BookCorpus and Wikipedia	0.81	-	0.62	**0.55**
BERT.avg	Wikipedia	0.812	-	0.594	0.468
ELMo.avg	Wikipedia	0.668	-	0.583	0.436
SG	Wikipedia	0.752	-	0.610	0.394
CBOW	Wikipedia	0.727	-	**0.627**	0.380
LSA	Wikipedia	0.681	-	0.614	-
HAL	Wikipedia	0.261	-	0.195	-
COALS	USENET	0.682	0.671	0.626	-
HAL	USENET	0.153	0.319	0.311	-
LSA	USENET	0.656	0.731	0.599	-

In table 6, we present our results on the 4 Word Similarity tests included in the experiment. We organized the table in three section on the base of the corpus used to train the model.

The results of other algorithms were taken from Rohde, Gonnerman, and Plaut (2006) for the models trained on the USENET Corpus (1.2 billion words); from Jurgens and Stevens (2010) for LSA and HAL trained on WIKI corpora (respectively 600 and 900 million words); and from Padó and Lapata (2007) for the DVS model.

The scores for Contextualized Models were collected from two sources: Lenci et al. (2022) analyzes three different types of BERT embeddings: BERT.F4 which uses the sum of the embeddings from the first four layers; BERT.L4 which uses the sum of the embeddings from the last four layers; and BERT.L which uses the embeddings from the last layer. In all cases, the authors used the bert-large-uncased model (pretrained on *BooKCorpus*[5] and English Wikipedia). We report only the model which records the best

[5] BookCorpus is a corpus of 11.038 unpublished books

scores (BERT.L4).
Wang, Cui, and Zhang (2021) adopts three different methods to use static similarities from BERT and ELMO, but we selected the one which obtained the best results (defined as BERT.avg and ELMo.avg by the authors). From the same paper, we also report the score of Skip-Gram (SG) and CBOW. All the models presented in Wang, Cui, and Zhang (2021) are trained on a Wikipedia Dump (1.1 billion tokens).

For COALS-BNC we used the *sspace package* and set the same parameters specified by the authors, 14,000 dimensions for each vector, 15,000-word vectors, and a list of *syntactic words* and punctuation excluded from the calculation of the matrix. For W2V-BNC we used the Python Gensim package[6], which uses CBOW as the default model, with automatic frequent phrases detection, a window-dimension of 5 and 200 dimensions.

In accordance with Rohde, Gonnerman, and Plaut (2006), we used the rank-order Correlation (Spearman's rho) to calculate the correlation between our results and human ratings, and we used the best-fit exponential scaling of similarity scores: scores of less than 0 are set to 0, and positive scores are replaced by $S(a, b)^t$ where $S(a, b)$ is the similarity score obtained, and t is an exponential that maximizes the model's correlation. A value of $t > 1$ increases sensitivity at the high end of the rating scale and $t < 1$ at the low end. We used a $t = 0.7$ for the SD-W2 model and W2V and 0.6 for COALS trained on BNC. The similarity values have been generated using Pearson's correlation for SD-W2-BNC and Cosine Similarity for the other models (including SD-W2-Wiki).

Concerning *SimLex-999*, we also followed the experiment conducted by Hill, Reichart, and Korhonen (2015) who tested their data set on a representative set of DS models such as LSA, VSM (Kiela and Clark 2014) or Word2Vec (Mikolov et al. 2013a). In table 7 we compare the correlation of both SD-W2 models with the correlation of LSA and W2V trained on the RCV1 Corpus (~ 150 million words) (Lewis et al. 2004) with two different window sizes (10 and 2) as reported by Hill, Reichart, and Korhonen (2015), and with COALS and Word2Vec trained on BNC.

Table 7
Comparison of SD-W2, COALS-BNC, W2V, and LSA on SimLex-999

Algorithm	SimLex-999	Most Associated 333	Adjectives (111)	Nouns (666)	Verbs (222)	Concrete (250)	Abstract (250)
SD-W2-Wiki	0.394	0.212	0.421	0.455	0.191	0.425	0.296
W2V-Wiki	0.414	0.260	-	-	-	-	-
SD-W2-BNC	0.303	0.107	0.413	0.359	0.08	0.315	0.227
COALS-BNC	0.220	0.017	0.338	0.253	0.034	0.212	0.200
W2V-BNC	0.324	0.057	0.463	0.342	0.170	0.339	0.369
LSA-RCV1 (2)	0.233	0.009	0.375	0.270	0.085	0.226	0.185
LSA-RCV1 (10)	0.238	0.070	0.272	0.298	0.008	0.325	0.209
W2V-RCV1 (2)	0.282	0.178	0.436	0.303	0.161	0.248	0.306
W2V-RCV1 (10)	0.266	0.176	0.406	0.278	0.114	0.236	0.309

Table 7 refers to different subsets of SimLex-999. The correlation for the whole set is shown in the second column. The third column reports the value of a subset of 333 most strongly associated concepts, according to the University of South Florida Free Association Database (USF) (Nelson, McEvoy, and Schreiber 2004). Association data were generated by human subjects who produced a set of associated words for 5000 concepts.

[6] `https://radimrehurek.com/gensim/models/word2vec.html`

5.1.1 Discussion

In table 6 we present the results of our model compared with many state-of-the-art models, in relation to corpora of different kinds and dimensions. This comparison allows us to study the importance of the corpus dimension and typology on the generation of co-occurrence values. Starting from the models trained on BNC, we must underline that the CBOW model of Word2Vec reaches higher results compared with SD-W2, for all four data-sets. As pointed out also by Hill, Reichart, and Korhonen (2015), SimLex-999 is *notably more challenging* than the other data-sets: nevertheless, the results of W2V trained on BNC also surpass the scores of the same model trained on RCV1 as reported by Hill, Reichart, and Korhonen (2015) and presented in table 7. Concerning the other data-sets, SD-W2 achieves better results than COALS and DVS, from which it draws inspiration and obtains similar results to W2V.

If the corpus dimension is increased, the results of our model become comparable to those of the contextualized models. Regarding the smaller data-sets, SD-W2 shows the best results with a precision of 0.842, overcoming both BERT (0.81) and ELMo (0.69), but also the two Mikolov models Skip-Gram and CBOW (respectively 0.75 and 0.73). With bigger data-sets such as Word-Sim353 and Sim-Lex999 the performance of SD-W2 decreases, but they are still comparable with the results of other models trained on a Wikipedia Corpus. In fact, for WS353, our results are in line with those of LSA, SG and ELMo and slightly lower than those of CBOW and BERT.L4. For SimLex999, the results of SD-W2 are similar to SG and CBOW but significantly lower than BERT and ELMo.

We tested SD-W2 also on the subsets of SimLex999 and compared the results with those presented by Hill, Reichart, and Korhonen (2015) and with the models trained on the BNC. In table 7, we present the results of SD-W2 compared with W2V, both trained on Wikipedia, but also the results of the same models trained on BNC. We also compared our model trained on BNC with LSA and W2V trained on RCV1 (similar in size to BNC).

The performance of our model varies according to subset and training corpus: if we consider the models trained on Wikipedia, we can compare SD-W2 only with W2V and only for the full data-set and the Most Associated 333 pairs. In this case, the results are very similar, especially with the full data-set. Regarding the models trained on the smaller corpora, if we consider the models presented in Hill, Reichart, and Korhonen (2015), we obtain high results over the whole Simlex, the Most Associated 333, and the subset of Nouns. We performed worst over the other subsets such as Verbs and Abstract Nouns.

5.2 Synonym Detection

Landauer and Dumais (1997) tested LSA on the *Test of English as a Foreign Language* (TOEFL) for the first time. In the paper, the TOEFL test was reduced to 80 questions (items) requiring the synonym of a given target word to be identified in a group of 4 words. The original test also provided a small clause context to the target word that Landauer had deleted in his computational experiment. After this test, many other tests have been used to evaluate DS models, such as the ESL (*English as a Second Language*) (Turney 2001) test or the *Reader's Digest Word Power test* (Jarmasz and Szpakowicz 2004). In particular, the ESL test consists of 50 items that tend to include words with higher frequencies than the TOEFL items. ESL items are based on a more subtle discrimination of meaning. For the target word *passage*, for example, the four alternatives are *hallway*, *ticket*, *entrance*, *room* and the solution is the word *hallway*.

In this paper, we will test our model on the TOEFL and ESL tests. The results are shown in Table 8[7].

Table 8
Comparison of different algorithms on TOEFL and ESL tests

Algorithm	Corpus	TOEFL	ESL
SD-W2	BNC	0.69	0.49
COALS	BNC	0.75	0.46
DVS	BNC	0.73	-
W2V	BNC	0.75	0.64
SD-W2	WIKI	0.76	0.54
BERT.L4	WIKI	0.89	0.60
HAL	WIKI	0.50	0.31
LSA	WIKI	0.61	0.54
COALS	USENET	0.86	0.52
HAL	USENET	0.56	0.26
LSA	USENET	0.53	0.43

In these experiments, we calculated the semantic similarity between the target word and each item's words. We took the word with the highest similarity value as the correct answer and then calculated the accuracy by counting the correct answers.

Considering a human average score of 64.5% for the TOEFL test, we can affirm that SS-W2 surpassed the human rating.

5.2.1 Discussion

The semantic similarity task tackled in this section includes two classic experiments: TOEFL and ESL. In comparing different models, the use of the same (training) corpus would have guaranteed consistent, better aligned results (Padó and Lapata 2007). Nevertheless, it would have been a major process to train a different model on BNC, so we must rely on the accuracy values reported in other papers. Table 11 shows the accuracy of the same DS models presented in the previous section, so we inserted only two scores achieved by models trained on the BNC corpus. Regarding the DVS model, we only have information on the TOEFL test because it is the only test the authors considered in their experiment.

According to the accuracy highlighted by Padó and Lapata (2007), we know that the PMI-IR model (Turney 2001) trained on BNC attains 61.3% accuracy, while the original model trained on a large Web-based corpus achieves 72.5%.

As for the similarity task, we report the results of our model trained on the two different corpora, BNC and Wikipedia. The results of our model are below expectations both for the one trained on BNC and for the one trained on Wikipedia and for both data-sets. If we do not consider the older models, SD-W2 obtains very low results for the two data-sets, reaching the best precision of 0.759 on TOEFL when trained on Wikipedia which is similar to the precision of the other model trained on a smaller

[7] A complete list of TOEFL results for DS models is shown on `https://aclweb.org/aclwiki/TOEFL\_Synonym\_Questions\_(State\_of\_the\_art)`

corpus. Although our model achieves the average precision score for count models (Baroni, Dinu, and Kruszewski 2014) on the TOEFL, we believe its precision to be too low when compared to the output of other models like COALS.

In conclusion, the results of SD-W2 are above the average of the tested models both for TOEFL and ESL. In Lapesa and Evert (2014) the authors claim that the parameters affecting the accuracy of the model for the TOEFL test are the distance metric, the score and the transformation. Cosine similarity, for example, produces better results than other metrics, while the association measures based on significance tests achieve the best results. Window-size might also affect model performance, the best results being achieved with a window-size of 2. Nevertheless, Lapesa and Evert (2017) tested the best parameters for dependency-based DSM, and the authors found that the parameters with a strong impact are metric, score and transformation. Analyzing the results of Lapesa and Evert (2017), we can impute the lack of precision of SD-W2 mainly to the absence of dimension reduction.

Also Bullinaria and Levy (2007) analyses the importance of different parameters on many semantic tasks. For the TOEFL task, for example, the models tested in the paper obtain the best results with small window-size. Since in our model the windows size do not correspond to a specific number of terms, we can't really control the number of words that belongs to the context of a given term and this can negatively affect the precision of SD-W2. Nevertheless, the conclusions of Bullinaria and Levy (2007) contrast with those of Lapesa and Evert (2014) regarding the dimension reduction.

5.3 Single-Word Priming

Inspired by Padó and Lapata (2007), we decided to also test the SD-W2 model on a simulation of semantic priming. This task is addressed in other studies (Lund and Burgess 1996; McDonald and Brew 2004) and entails the exposure of semantic similarity or dissimilarity between words. According to Padó and Lapata (2007, p. 180), "if dependency-based models indeed represent more linguistic knowledge, they should be able to model semantic priming better than traditional word-based models".

The experiment is based on the Hodgson (1991) single-word priming study. The underlying principle is that the presentation of a *prime* word like *clown* could facilitate the lexical decision on a *target* word like *circus*. Hodgson proposed an experiment in which the human subjects must take a decision about 144 pairs of words belonging to six different lexical relations: synonymy (*trash-garbage*), superordination or subordination (*fuel-gas*), category coordination (*rectangle-circle*), antonymy (*enter-exit*), conceptual association (*clown-circus*), and phrasal association (*foreign-language*). The goal of the experiment was to investigate the influence of each lexical relation on the prime effect. The paired words were selected from different POS (Nouns, Verbs, and Adjectives) and represented an unambiguous example of the relation type. The results of the original experiment demonstrate that there is an equivalent priming effect for the six lexical relations.

This experiment was used in McDonald and Brew (2004) to test the ICE (*Incremental Construction of Semantic Expectations*) model. In Padó and Lapata (2007) the 143 original pairs (one synonymy pair was lost) were reduced by deleting pairs with at least one low-frequency word. The authors set the Lexical Relation and prime (related, unrelated) as independent variables. The dependent variable representing the quantity being measured is the semantic distance between the prime and the target. The distance between Related and Unrelated prime-target pairs simulates the priming effect. Since the Unrelated primes were not provided in the description of the original experiment, both DVS and ICE models used the averaged distance of a target to all other primes of the same relation as unrelated primes.

In order to measure the prime effect and compare the results with the DVS model, we performed a two-way analysis of variance (ANOVA) on the data generated by SD-W2, COALS-BNC and W2V-BNC. Lexical Relation (six levels) and prime (two levels) were the factors. SD-W2 showed a strong prime effect as with BNC (F(1,135) = 257.64, MSE = 2.15, $p < 0.01$) as with Wackypedia (F(1,135) = 435.64, MSE = 3.52, $p < 0.01$). The value of p is significant (< 0.01) and indicates a significant difference between Related and Unrelated pairs. Also COALS-BNC (F(1,135) = 163.92, MSE = 1.02, $p < 0.01$) and W2V-BNC (F(1,135) = 447.09, MSE = 10.05, $p < 0.01$) showed a significant prime effect.

Having determined that there are differences between Related and Unrelated prime targets, we need to quantify the magnitude of the prime effect. Padó and Lapata suggest using the Eta-squared (η^2) measure, often employed to calculate the strength of an experimental effect. The formula of Eta-squared is $\eta^2 = \frac{SS_{effect}}{SS_{total}}$, where SS_{effect} represents the variance (sum of square) created by one particular effect (the prime) and SS_{total} is the sum of the variance of all observations. It represents how the variability in the distance variable can be explained by priming (Related-Unrelated). DVS reports an eta^2 of 0.332. This means that DVS accounts for 33.2% of the variance. The eta^2 of SD-W2 trained on BNC is 0.477, while trained on Wackypedia is 0.566. COALS obtains 0.383. The eta^2 obtained by W2V-BNC is 0.613.

In order to verify the prime effect over all six relations, we produced different ANOVAs for each Lexical Relation. Table 9 reports the mean distance values for each relation in the Related and Unrelated condition. It also indicates the prime effect size for each relation for SD-W2, COALS-BNC, and DVS, calculated as Related-Unrelated.

Table 9
Mean distance values for the six Lexical Relations; Prime Effect size for SD-W2, COALS, DVS and W2V

Lexical Relation	Related	Unrelated	SD-W2 BNC Effect	SD-W2 WIKI Effect	COALS Effect	DVS Effect	W2V Effect
Synonymy	0.374391	0.141128	0.233262	0.294304	0.163129	0.165	0.514
Superordination	0.327209	0.126888	0.200321	0.287032	0.111652	0.106	0.386
Category coordination	0.340998	0.142409	0.198589	0.302349	0.124305	0.137	0.336
Antonymy	0.291833	0.142169	0.149664	0.197816	0.126387	0.165	0.409
Conceptual association	0.291064	0.122289	0.168775	0.172834	0.114011	0.083	0.404
Phrasal association	0.253054	0.125435	0.127619	0.132093	0.102564	0.043	0.282

5.3.1 Discussion

According to Padó and Lapata, the semantic priming must be modeled better by means of a model that can represent more linguistic knowledge. With this experiment, we point out that SD-W2 can show a reliable prime effect on the Hodgson experiment, surpassing the results of the other models tested on the same data set and trained with the same corpus. The significantly better results reached by Word2Vec reflect the advances of the DS models in the last years. The use of Neural Networks helps to produce better results although the corpus used was the same than other models.

Analyzing each Lexical Relation result presented in table 9, we observe a reliable prime effect on the six types for SD-W2. In particular, the model shows the best results with Synonymy and Superordination-subordination pairs (almost double the value obtained by COALS and DVS). Phrasal association, Conceptual association, and Category coordination obtain decent results compared with the DVS model but similar to COALS. As for Antonymy, SD-W2 shows the worst prime effect.

Analyzing the similarity generated by single pairs, we notice that the Antonymy relation shows no critical issues but the closest Related-Unrelated values. In phrasal pairs, on the other hand, there is a general greater deviation between Related and Unrelated similarities, although in three cases the Unrelated value is higher than the Related one (*help-wanted*, *mountain-range*, and *pony-express*). While two of these values are very close, the value of the pair *pony-express* is considerably lower than the average distances of all the other primes. The low value obtained by Phrasal association pairs can be attributed to the nature of this association. In effect, it depends on *in-praesentia* relations and is strongly influenced by the co-occurrence of the two pair words in the corpus. For example, the words *pony* and *express* have high frequencies in BNC, but the sequence *pony express* only appears twice. Contrariwise, in Wikipedia, there are many pages in which the two words appears in association (movies, tv shows,sports and other categories).

5.4 Operator-Argument selection

In section 2 we stated that, according to Harris's distributional hypothesis, the context selection of DS models must include not the graphical context of a target word but its syntactic context since, according to Harris's theory, the distribution of a word must be associated with the relation between Operators and Arguments. This kind of relationship is a syntactic relationship and can be brought out by a dependency tree. This is why the SD-W2 model relies on syntactical dependency and selects all the words included within a syntactic distance range as contexts of the target word.
In order to test the ability of our model to detect Operator-Argument relations, we set up a new experiment in which the model must connect a class of nouns with the verb form that selects this class as a right or left argument. For the vast majority of verbs, subject or object selection includes very generic classes of nouns. The verb *to sleep*, for example, selects animate entities (*the dog*, *the child*, *John*, etc.) as likely subjects, like many other verbs. A transitive verb such as to listen presents a similar distribution to sleep for the subject and a huge selection of nouns as the object.

For the *Operator-Argument selection test*, we needed a set of verbs whose distribution must be restrictive. A verb like *to smoke*, for example, includes the very restricted class of "smokable items" as the object. The word *cigarette* can be selected as the argument in a wide range of verbs with variable likelihood. Whereas, if we consider the information that the Operator and the Argument mutually exchange, we must find a stronger similarity between the noun and the verb *to smoke*. Following this hypothesis, we built a data set of verbs with restricted arguments.

This data set is based on the syntactic classes of verbs collected by the *Lexicon-Grammar Theory* (LG). LG, which is deeply connected to the Operators-Arguments theory, determines the structure of a large number of verbs (Gross 1975) that were classified on the basis of their shared syntactic features. Since there are only specific LG tables of English verbs (mainly phrasal verbs), we relied on the Italian classification (Elia 1984; Vietri 2004) from which we selected two classes of verbs with restricted arguments: class 2B and class 20R.

Thanks to this classification, we were able to extract, for example, all the intransitive verbs with one restricted argument (*to bark*, *to derail*, *to erupt*, etc.) from class 2B, or transitive verbs with restricted objects (*to smoke*, *to drink*, *to celebrate*) from class 20R.

Class 20R includes 77 verbal uses characterized by a syntactic structure of the kind $N_0VN_{1restricted}$. The verbs of class 20R present only one complement (direct object) which is strongly restricted to one or a specific class of objects. We select 25 verbs from this class which present a very restricted selection and are not ambiguous or used metaphorically.
Class 2B includes 45 intransitive verbs with a structure $N_{0restricted}V$. As for class 20R, the subjects present a selection of nouns restricted to one specific class. Likewise in this class, many verbs used metaphorically have been discarded.

Hence, 70 Italian verbs were selected. These verbs were then translated and the 68 which keep the same properties in both languages were selected. From the list of 68 English verbs, we selected a restricted group by deleting verbs that feature a restricted argument only in one interpretation (*to quote*, *to cultivate*), with very low frequencies (*to erupt*, *to engrave*, *to rebind*), and with a metaphorical use (*to roar*, *to shine*)

The final list included 26 verbs that were used to generate sets of 4 nouns, which can figure as the restricted subject or the restricted object of these verbs. The groups include nouns that must represent both prototypes of the class of nouns required by the verb and more peripheral nouns, with the least possible ambiguity. The nouns of one group may occasionally appear again in another group.

We decided to include some verbs with a very similar distribution, such as *cook* and *fry*, and test the models with subtler differences.

The problem of choosing a group of nouns that work as the subjects or objects of a given verb primarily applies to verbs with similar meanings. When we selected the group of nouns for the verb *to wear*, we freely selected nouns from the list of clothes. In fact, clothes represent the restricted distribution of objects for *to wear*. However, using a frequency criterion for representativeness, we look for the most representative and distinctive objects among the nouns of clothes (*shirt*, *hat*, *jeans* and *shoes*).

On the other hand, when choosing the nouns for *fry* or *cook*, which both select the same class of nouns (edible items or foods), we attempted to choose nouns that emphasise the variations between the two distributions. For *to fry* we selected *potatoes*, *chips*, *eggs* or *bacon*. Since *fry* can be considered as a subclass of *cook*, the latter can also select all those elements, but with less probability than *bean*, *pasta*, *rice* and *bread*.

The groups of nouns were submitted to 40 human subjects to test their capacity to connect the arguments with the correct verb. The subjects were Italian undergraduates and master's degree or PhD students with good linguistic skills. They were asked to read the list of verbs and, for each group of nouns, choose a verb that can select all four nouns in the group as subject or object.

We calculate the precision as the number of correct answers (verbs correctly associated with the list of nouns they select) divided by the total number of questions (26). The human subjects had issues with classes that can select very similar items such as *cook* and *fry* or *smell*, or *hunt*, *growl*, and *bark*, but in general, the average human precision is 0.923. This result validated the proposed group of nouns related to each verb: while most human subjects correctly associated nouns and verbs, some of them reported a precision range of 0.85 to 0.90. Only one subject scored 0.77. The fact that the human subjects confused *cook* with *smell*, which includes the nouns *flower* and *perfume*, or *hunt* with *bark*, which includes the noun *puppy*, indicates that many errors can be attributed to a cursory reading of the data.

Table 10 shows the selected verbs and the group of nouns.

Table 10
Data set for the Operator-Argument Selection Test

Verbs	Groups of selected nouns
fly	plane, robin, bird, helicopter
cook	bean, pasta, rice, bread
fry	potato, chips, egg, bacon
harvest	cereals, wheat, corn, grain
blossom	rose, violet, lily, daisy
growl	dog, monster, wolf, hound
gallop	rider, horse, pony, deer
asphalt	street, ground, square, road
boil	soup, water, milk, bean
hunt	fox, deer, elephant, bird
wear	shirt, hat, jeans, shoes
celebrate	marriage, wedding, festival, christmas
smoke	cigarette, cigar, tobacco, weed
drink	water, milk, whisky, juice
prune	pine, tree, oak, branch
prescribe	drug, medicine, pill, treatment
print	newspaper, book, picture, photo
drive	car, bus, train, truck
shear	hair, sheep, fur, goat
smell	garlic, cheese, flower, perfume
play	football, role, tennis, guitar
sing	song, carol, prayer, hymn
run	championship, race, marathon, tender
abort	baby, male, children, pregnancy
bark	dog, puppy, wolf, hound
bellow	bull, cow, elephant, ox

We tested the SD-W2 model with this data set by computing the best candidate verb for a group of nouns as the one with the highest average semantic distance from every noun. The precision of the SD-W2 model was 0.73 while COALS obtained 0.57. Word2Vec reaches a precision of 0.808.

Table 11
Comparison of different algorithms on Verb Selection Test

Algorithm	Corpus	Precision
SD-W2	BNC	0.73
COALS	BNC	0.57
W2V	BNC	0.81
SD-W2	WIKI	0.61
W2V	WIKI+GIGAWORD	0.65

As shown in table 11, we also tested SD-W2 model trained on Wackypedia and the GLOVE Word2Vec pre-trained model (6 billion of words from Wikipedia and Gigaword English Corpus) with 200 dimensions. Interestingly, those two versions, trained on larger corpora, obtain the worst results, underlining that the precision of the model, for this task, is not influenced by the corpus dimension, but by its content.

5.4.1 Discussion

With this experiment, we aimed to test the SD-W2 model's ability to detect the connection between a verb and the class of noun it selects as an argument. As for the Semantic Priming experiment, we think that dependency models must model this kind of relationship better because they explore the syntactic connection between words. Our experiment reveals two critical weaknesses: first, we compare our model only with COALS and Word2Vec; second, the data set is still incomplete and needs to be improved and tested by more human subjects.

In actual fact, we can only study the results of the SD-W2 model by exploring the critical issues we identified. The best model configuration (BNC) fails in the classification of seven groups: it confuses *fry* with *cook*, *asphalt* with *drive*, *prune* with *bark*, *shear* with *wear*, *abort* with *fly*, *bark* with *growl* and *bellow* with *fly*.

The model which reaches the best results was Word2Vec, which share some errors with SD-W2 (*prune,shear,abort* and *bellow*) but also confuses *run* with *gallop*.

In some cases, we expected the model to make the error, such as in the case of *to bark* and *to growl* which have a very similar meaning and select a similar group of items. The same goes for *to cook* and *to fry*.

As for *to prune* and *to bark*, we must attribute the error to the ambiguity of *bark*, which can also mean *the tough protective outer sheath of a tree trunk*. Since we train the model on a lemmatized corpus, we must use the dictionary form of the verbs, and we cannot disambiguate the meaning by using, for example, the past tense. This hypothesis is also confirmed by the error of Word2Vec.

The case of *to asphalt* and *to drive* is also clear, because for the latter verb what interferes may be a locative complement. In fact, drive has a higher similarity with *road* or *street*, much more than the similarity between the two words and *asphalt*.

With the verbs *abort* and *bellow*, the model confuses them with *fly*. In the first case, the word abort in BNC seems to be connected with the domain of computer science (as in he *terminates/aborts the program/process*) and it manifests a weak semantic association with all the words in the group. On the other hand, *fly* has higher similarity values with *baby* and *male* which are also related to the sphere of zoology. The word *male*, for example, has a strong association with the word *bird*.

The word *bellow* obtains similarity values with the four words in the group comparable to the ones obtained by *fly*, but the latter has a higher value with all the words. We observe the same behaviours in Word2Vec results for the two group of words.

In order to visualize the neighborhood of a verb like *to fly* or *to shear*, we developed a network composed of three levels of the verb's neighbors: we extracted the verb's 50 nearest objects (first-level objects) and their similarity scores, and performed a 10-object extraction (second-level objects) for each of the 50 first-level objects. We then replicated the same process for the second-level objects (third-level objects).

We generated a network in which the nodes are words and the weighted edges are similarity scores. We used Gephi (Bastian, Heymann, and Jacomy 2009) to build up the visualization and performed two specific graph algorithms. First, we calculated the degree of each node to point out words that frequently appear as the verb's nearest neighbors; second, we ran the *Modularity Class* algorithm to calculate sub-communities of nodes and easily identify specific classes of words. Modularity Class (Blondel et al. 2008; Lambiotte, Delvenne, and Barahona 2008) was applied to the network with a resolution parameter of 2 to minimize the number of generated classes.

An example of the network is reported in figure 2, which shows two-word networks: the upper figure represents the neighborhood of the word *shear* (the yellow node). As can be seen, the words that emerge are all related to the domain of physics. The noun *shear* represents "a movement in the plates in the surface of the earth that causes them to change shape or break" and the verb *to shear* also refers to a deformation of a material substance in which parallel internal surfaces slide past one another.

The figure below refers to the word *fly* and shows the relation of the verb with its possible subjects. The Modularity Class identifies a class of *animals* (red nodes) in which the word *bird* stands out, but also a class of *vehicles* (green nodes), *places* (blue nodes), and *motion verbs* (black nodes).

The differences between a verb associated with the correct group of nouns (*fly*) and a verb where the system produces an error (*shear*) emerge clearly in this kind of visualization. In fact, in the network of *shear*, there is no sign of the nouns in the corresponding group. This is confirmed by figure 3 which contains the network of the word *celebrate*.

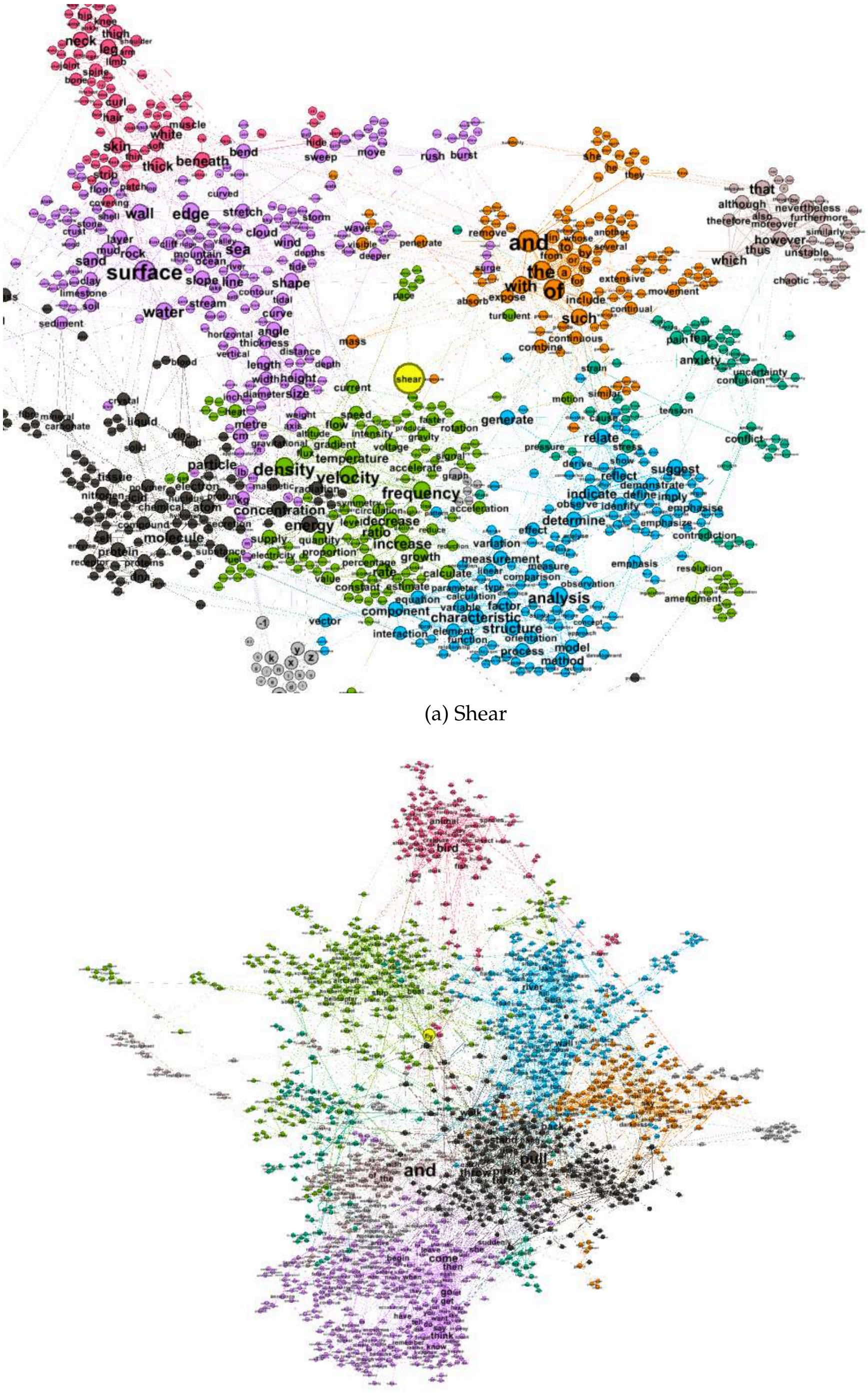
Among the neighbors of *celebrate* we find a group of words related to the temporal dimension (*weekend*, *day*, *evening*), music or arts in general (*concert*, *exhibition*), and events (*ceremony*, *festival*, *protest*).
This indicates that, in some cases, the problem may lie in the corpus where a specific meaning of a word is privileged and not in the model.

In general, SD-W2 obtained good results, compared to COALS-BNC. An analysis of the errors of our model points out that the words in the group associated with *to bark* and the words in the group associated with *to fry* belong to the same category of respectively Animals and Food, which are also selected by *to cook* and *to growl*. Even if a human subject can detect differences between these groups, we can consider this model's errors as minor. If these two groups of words are excluded from the data set or if the two automatic evaluations are considered exact, SD-W2 exceeds 80% accuracy, surpassing the score obtained by some of the human subjects who took part in the experiment.

6. Conclusion

In this paper, we have presented a new model for Distributional Semantics. The model, called SD-W2, uses syntactic distances extracted from a parsed text to build a word's context. From the distributional hypothesis analysis conducted, we argue that the context of which Harris speaks is syntactic because every analysis of the meaning must be based on the Operator-Argument relation. To base our distributional analysis on the syntactic dependencies between words, we use a model that propagates the influence of a target word on its related words at a specific syntactic distance. In order to calculate this influence, we tested a linear method in which each word directly connected with the target obtains a higher value, and this value is decreased by 1 for more distant words. We also tested a different methodology in which we calculated the weight of the influence of the target word over the other words as a function of the percentage of the sum of its degrees divided by the distance.

Since we obtained the best results with the second methodology, we tested the model with the latter weight function in three experiments used by many other authors. The first family of experiments concerns semantic similarity. The model must replicate



(a) Shear

(b) Fly

Figure 2
Gephi network for two words (*shear* and *fly*); different colours correspond to different classes. Text size depends on the node degree.

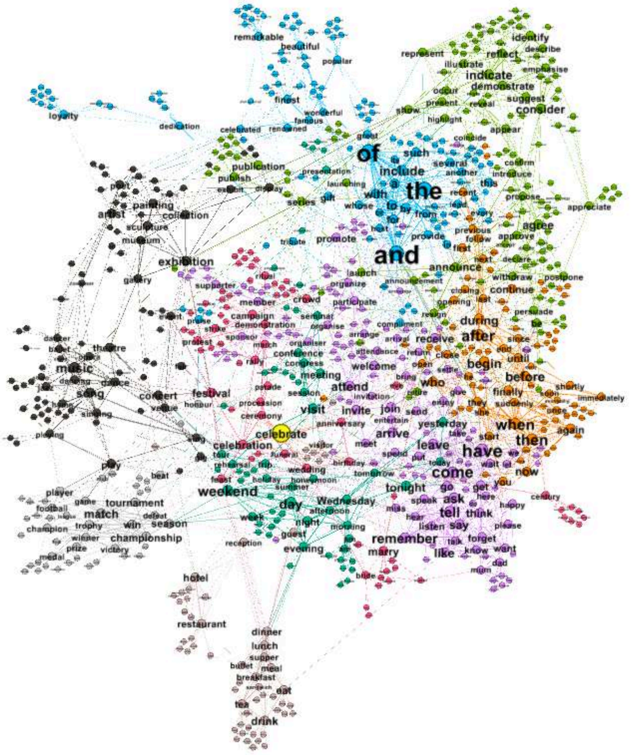


Figure 3
The network of *celebrate*

human judgment on the similarity between pairs of words. The performance of SD-W2 in this experiment exceeds the average of compared models and some cases are above the performance of the predict or contextualized models.

The second family of experiments deals with synonymy and is based on synonymy tests for foreigners. In this case, the results of SD-W2 are above the average of compared models. This result can be imputed to the lack of matrix dimensionality reduction, which is a parameter of crucial importance in this class of experiments. In the future, we plan to add an SVD reduction algorithm to the SD-W2 model to verify this assumption.

The third experiment regards single-word priming. The test is modeled over the Hodgson experiment (Hodgson 1991) and calculates the model's ability to simulate the prime effect on six different kinds of lexical relations. The results of SD-W2 are very encouraging for this task, surpassing those obtained for the compared models, except for Word2Vec which trained on BNC and obtained the best performance.

Finally, we subjected the model to a new experimental test regarding the operators' argument selection. Given a set of groups of four nouns belonging to a specific semantic class, the model must calculate the verb that selects the words of each group as an argument. The model reached a good accuracy score (73%) with errors that, in many cases, are due to the verb's ambiguity. Testing two models (SD-W2 and W2V) trained on different larger corpora, we realized that this parameter is not relevant to the task. The BNC corpus obtained the best results despite its reduced dimension. In the future we plan to enlarge the experiment and test our model trained on different corpora in order to define the parameters that achieve the best results for the task.

We demonstrate that a dependency model could achieve good results without a large and expensive pre-processing phase. Comparing our model with a similar word-window model like COALS, trained on BNC, we demonstrate that SD-W2 can surpass COALS in almost all the selected tasks and with a comparable amount of pre-processing. Consequently, we demonstrate that growth in corpus size results in the exponential improvement in our model's performance. Training the model on a large corpus such as Wackypedia, its performance reaches the performance levels of DL-Based models in some cases.

References

Audet, Chad and Curt Burgess. 1999. Using a high-dimensional memory model to evaluate the properties of abstract and concrete words. In *Proceedings of the Twenty-first Annual Conference of the Cognitive Science Society*, pages 37–42, Vancouver, Canada, December. Citeseer.

Azzopardi, Leif, Mark Girolami, and Malcolm Crowe. 2005. Probabilistic hyperspace analogue to language. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 575–576, Salvador, Brazil, August. ACM.

Baroni, Marco, Raffaella Bernardi, and Roberto Zamparelli. 2014. Frege in space: A program for compositional distributional semantics. *Linguistic Issues in language technology*, 9(6):5–110.

Baroni, Marco, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43(3):209–226.

Baroni, Marco, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, United States, June.

Baroni, Marco and Alessandro Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721.

Bastian, Mathieu, Sebastien Heymann, and Mathieu Jacomy. 2009. Gephi: an open source software for exploring and manipulating networks. In *Proceedings of the Third International AAAI Conference on Weblogs and Social Media*, volume 3, pages 361–362, San Jose, California, May.

Blondel, Vincent D., Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):1–12.

Boleda, Gemma. 2020. Distributional semantics and linguistic theory. *Annual Review of Linguistics*, 6:213–234.

Buchholz, Sabine and Erwin Marsi. 2006. CoNLL-X shared task on multilingual dependency parsing. In *Proceedings of the tenth conference on computational natural language learning (CoNLL-X)*, pages 149–164, New York City, United States, June. ACL.

Bullinaria, John A. and Joseph P. Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior research methods*, 39(3):510–526.

Burgess, Curt. 1998. From simple associations to the building blocks of language: Modeling meaning in memory with the HAL model. *Behavior Research Methods, Instruments, & Computers*, 30(2):188–198.

Burgess, Curt. 2001. Representing and resolving semantic ambiguity: A contribution from high-dimensional memory modeling. In *On the consequences of meaning selection: Perspectives on resolving lexical ambiguity*. American Psychological Association Washington, DC, pages 233–260.

Chersoni, Emmanuele, Enrico Santus, Philippe Blache, and Alessandro Lenci. 2017. Is structure necessary for modeling argument expectations in distributional semantics? In *IWCS 2017 - 12th International Conference on Computational Semantics - Long papers*, Montpellier, France, September.

Chersoni, Emmanuele, Enrico Santus, Chu-Ren Huang, and Alessandro Lenci. 2021. Decoding word embeddings with brain-based semantic features. *Computational Linguistics*, 47(3):663–698.

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

Elia, Annibale. 1984. *Le verbe italien: les complétives dans les phrases à un complément*. Schena; Nizet.

Finkelstein, Lev, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th international conference on World Wide Web*, pages 406–414, New York, United States, May.

Gale, William A., Kenneth W. Church, and David Yarowsky. 1995. Discrimination decisions for 100,000-dimensional spaces. *Annals of Operations Research*, 55(2):323–344.

Grefenstette, Gregory. 1992. Sextant: Exploring unexplored contexts for semantic extraction from syntactic analysis. In *30th Annual Meeting of the Association for Computational Linguistics*, pages 324–326, Newark, United States, June.

Grefenstette, Gregory. 1994. Corpus-derived first, second, and third-order word affinities. In *Proceedings of the 6th International Congress on Lexicography (EURALEX 1994)*, pages 279–290, Amsterdam, Netherlands, August. Rank Xerox Research Centre.

Gross, Maurice. 1975. *Méthodes en syntaxe: régime des constructions complétives*, volume 1365. Hermann Paris.

Harris, Zellig. 1968. *Mathematical structures of language*, volume 21. Interscience, New York, United States.

Harris, Zellig. 1976a. On a theory of language. *The Journal of Philosophy*, 73(10):253–276.

Harris, Zellig. 1976b. A theory of language structure. *American Philosophical Quarterly*, 13(4):237–255.

Harris, Zellig. 1988. *Language and information*. Columbia University Press, New York, United States.

Harris, Zellig. 1991. *Theory of language and information: a mathematical approach*. Oxford University Press, Oxford, UK.

Harris, Zellig S. 1946. From morpheme to utterance. *Language*, 22(3):161–183.

Harris, Zellig S. 1952. Discourse analysis. *Language*, 28(1):1–30.

Harris, Zellig S. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Harris, Zellig S. 1957. Co-occurrence and transformation in linguistic structure. *Language*, 33(3):283–340.

Hill, Felix, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Hodgson, James M. 1991. Informational constraints on pre-lexical priming. *Language and Cognitive Processes*, 6(3):169–205.

Jarmasz, Mario and Stan Szpakowicz. 2004. Roget's thesaurus and semantic similarity. In Nicolas Nicolov, Kalina Bontcheva, Galia Angelova, and Ruslan Mitkov, editors, *Recent Advances in Natural Language Processing III: Selected Papers from RANLP 2003*. Benjamins, pages 111–120.

Jurgens, David and Keith Stevens. 2010. The S-Space package: An open source package for word space models. In *Proceedings of the ACL 2010 System Demonstrations*, pages 30–35, Uppsala, Sweden, July.

Kanerva, Pentii, Jan Kristoferson, and Anders Holst. 2000. Random indexing of text samples for latent semantic analysis. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, volume 22, Philadelphia, United States, August.

Kiela, Douwe and Stephen Clark. 2014. A systematic study of semantic vector space model parameters. In *Proceedings of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 21–30, Gothenburg, Sweden, April.

Lambiotte, Renaud, J-C Delvenne, and Mauricio Barahona. 2008. Laplacian dynamics and multiscale modular structure in networks. *arXiv preprint arXiv:0812.1770*.

Landauer, Thomas K. and Susan T. Dumais. 1997. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211.

Lapesa, Gabriella and Stefan Evert. 2014. A large scale evaluation of distributional semantic models: Parameters, interactions and model selection. *Transactions of the Association for Computational Linguistics*, 2:531–546.

Lapesa, Gabriella and Stefan Evert. 2017. Large-scale evaluation of dependency-based DSMs: Are they worth the effort? In *Proceedings of the 15th Conference of the European Chapter of the*

Association for Computational Linguistics: Volume 2, Short Papers, pages 394–400, Valencia, Spain, April.

Leech, Geoffrey Neil. 1992. 100 million words of English: the British National Corpus (BNC). *Language Research*, 28(1):1–13.

Lenci, Alessandro. 2008. Distributional semantics in linguistic and cognitive research. *Italian journal of linguistics*, 20(1):1–31.

Lenci, Alessandro. 2018. Distributional models of word meaning. *Annual review of Linguistics*, 4:151–171.

Lenci, Alessandro, Magnus Sahlgren, Patrick Jeuniaux, Amaru Cuba Gyllensten, and Martina Miliani. 2022. A comparative evaluation and analysis of three generations of distributional semantic models. *Language Resources and Evaluation*, 56:1269–1313.

Levy, Omer and Yoav Goldberg. 2014. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, United States, June.

Lewis, David D., Yiming Yang, Tony G. Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397.

Lin, Dekang. 1997. Using syntactic dependency as local context to resolve word sense ambiguity. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 64–71, Madrid, Spain, July.

Liu, Haitao, Chunshan Xu, and Junying Liang. 2017. Dependency distance: A new perspective on syntactic patterns in natural languages. *Physics of life reviews*, 21:171–193.

Lund, Kevin and Curt Burgess. 1996. Producing high-dimensional semantic spaces from lexical co-occurrence. *Behavior Research Methods, Instruments, & Computers*, 28(2):203–208.

Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, Baltimore, United States, June.

McDonald, Scott and Chris Brew. 2004. A distributional model of semantic context effects in lexical processing. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 17, Barcelona, Spain, July.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013*, Scottsdale, United States.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26:3111–3119.

Miller, George A. and Walter G. Charles. 1991. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28.

Nelson, Douglas L., Cathy L. McEvoy, and Thomas A. Schreiber. 2004. The university of south florida free association, rhyme, and word fragment norms. *Behavior Research Methods, Instruments, & Computers*, 36(3):402–407.

Nivre, Joakim, Johan Hall, and Jens Nilsson. 2006. Maltparser: A data-driven parser-generator for dependency parsing. In *Proceedings of the 5th International Conference on Language Resourcers and Evalutation (LREC 2006)*, volume 6, pages 2216–2219, Genova, Italy, May.

Padó, Sebastian and Mirella Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199.

Peters, Matthew E, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, United States, June.

Rohde, Douglas, Laura Gonnerman, and David Plaut. 2006. An improved method for deriving word meaning from lexical co-occurrence. *Communication of the ACM*, 8(01).

Rohde, Douglas L.T. 2002. Methods for binary multidimensional scaling. *Neural Computation*, 14(5):1195–1232.

Rubenstein, Herbert and John B. Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.

Ruge, Gerda. 1992. Experiments on linguistically-based term associations. *Information Processing & Management*, 28(3):317–332.

Sahlgren, Magnus. 2005. An introduction to random indexing. In *Proceedings of Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering*, Copenhagen, Denmark, August.

Sahlgren, Magnus. 2008. The distributional hypothesis. *Italian Journal of Disability Studies*, 20:33–53.

Schober, Patrick, Christa Boer, and Lothar A. Schwarte. 2018. Correlation coefficients: Appropriate use and interpretation. *Anesthesia & Analgesia*, 126(5):1763–1768.

Schutze, Hinrich. 1992a. Dimensions of meaning. In *Supercomputing'92: Proceedings of the 1992 ACM/IEEE Conference on Supercomputing*, pages 787–796, Minneapolis, United States, November. IEEE.

Schütze, Hinrich. 1992b. Word space. In *Advances in Neural Information Processing Systems (NIPS Conference)*, volume 5, pages 895–902, Denver, United States. Morgan-Kaufmann.

Schutze, Hinrich and Jan O. Pedersen. 1995. Information retrieval based on word senses. In *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, Las Vegas, United States, April. Citeseer.

Strzalkowski, Tomek. 1994. Building a lexical domain map from text corpora. In *COLING 1994 Volume 1: The 15th International Conference on Computational Linguistics*, Kyoto, Japan, August.

Taieb, Mohamed Ali Hadj, Torsten Zesch, and Mohamed Ben Aouicha. 2020. A survey of semantic relatedness evaluation datasets and procedures. *Artificial Intelligence Review*, 53(6):4407–4448.

Turney, Peter D. 2001. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *Proceedings of Machine Learning: ECML 2001*, pages 491–502, Freiburg, Germany, September. Springer.

Vietri, Simona. 2004. *Lessico-grammatica dell'italiano. Metodi, descrizioni e applicazioni*. Utet, Torino.

Wang, Yile, Leyang Cui, and Yue Zhang. 2021. Improving skip-gram embeddings using BERT. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:1318–1328.

Yarowsky, David. 1992. Word-sense disambiguation using statistical models of roget's categories trained on large corpora. In *COLING 1992 Volume 2: The 15th International Conference on Computational Linguistics*, Nantes, France, August.