Università degli Studi di Padova
Dipartimento di Matematica "Tullio Levi-Civita"

Corso di Laurea Magistrale in Matematica

Bias Analysis in Word Embeddings with Alignment Techniques

Relatore:
Giovanni Da San Martino

Correlatore:
Francesco Rinaldi

Laureanda:
Elena Della Casa
Matricola 2017686

23 Febbraio 2023

Contents

Abstract

In the field of Natural Language Processing, word embeddings are fundamental tools to represent the semantic relations among words. These tools are built by training learning algorithms on large corpora of textual data, which often reflect different types of biases and cultural peculiarities inherited by the society itself. Since word embeddings are the state-of-the-art representations in NLP tasks, biases are likely to be carried over by Machine Learning algorithms, which may, in turn, reinforce them. The present work leverages sparse optimization techniques to find a transformation among word embeddings trained on different corpora, able to highlight different types of biases in the data. Moreover, this study attempts to analyze the transformed data in order to detect the presence of cultural differences, both known and unknown.

Introduction

Natural language processing (NLP) is a sub-field of artificial intelligence that is concerned with how computers can process, understand and generate natural language ((Kaddari et al., 2021)). We could consider the adjective *natural* as opposed to *artificial*, a characteristic proper of symbolic languages of logic and mathematics, whose sintax is given by explicit and agreed rules ((Picardi, 1999)). On the other hand, natural language is full of contradictions, ambiguities, exceptions and it is not always logical. We could say that the ambitious aim of NLP is indeed to explain natural language by using artificial language.
Most Natural Language Processing (NLP) algorithms nowadays rely on vectorial representations of words, called ***word embeddings***. It is well known that such fundamental representations inherit cultural and social traits from data they are trained on, giving rise to various problematic situations. First, the distortion of word embeddings due to cultural peculiarities of the specific textual data might lead to various types of error in NLP tasks. Second, dangerous prejudice can be propagated and amplified by word embeddings, employed by systems massively used by millions of users in everyday life, influencing their perception of the world.
The goal of the present thesis is to provide a method to automatically detect when and how cultural influence hidden in data affects the inner structure of words embeddings, causing a phenomenon we call **cultural semantic conditioning**. Our method is based on ***alignment***, a map between two different embedding spaces, applied by multiplying each vector of the source space by a transformation matrix $\mathbf{W}$. By the resolution of nonlinear optimization problems, we compute different transformation matrices, which should place the images of word vectors more distant to the target vector if they are more likely to be subject to cultural semantic conditioning, and viceversa. By applying such new transformation, we manage to connect embeddings trained on different textual corpora, and then to quantify the cultural semantic conditioning over all its common words by using different measures we propose in this work.

The thesis is structured into two parts: part I consists of a general overview of the theory we rely on in our research, while the part II illustrates our contributions and the results of our experiments. More specifically, part I is made up by three chapters: in the chapter 1 we explore word embeddings, their properties, and popular models used to generate them; in the chapter 2, we overview linear and nonlinear optimization models, focusing on the FW's method in the end; in chapter 3 we investigate the phenomenon of cultural semantic conditioning, relying on renowned research works. Then, the part II is divided into chapter 4, which delves into our idea and our new contributions, and chapter 5, containing the results of our experiments.

Part I

The Backround

Chapter 1

Word Embeddings: A General Overview

The term word embedding denotes the vectorial representation of words widely used in the field of NLP, made up by particularly dense vectors. Mathematically, it can be seen as a mapping e from a vocabulary $\mathcal{D} = \{w_i\}_{i=i,\dots N}$, a set of N words, to a d-dimensional vectorial space $\mathcal{V} = \{\vec{w_i}\}_{i=i,\dots N}$ ((Jurafsky and Martin, 2000)).

$$e : \mathcal{D} \xrightarrow[w \mapsto e(w) = \vec{w}]{} \mathcal{V}$$

As a result, $\mathcal{V} \cong \mathbb{R}^d$, where d can take different values: the most common are 50, 100, 200, 300. Usually, $\mathcal{V} \cong \mathbb{R}^{300}$, since 300 is the most commonly used dimensionality in various studies ((Yin and Shen, 2018)). Word embeddings are built on vector semantics.

1.1 Vector Semantics and the Distributional Hypothesis

The distributional hypothesis Vector semantics is the model used to represent words meaning in NLP, and it is based on the **distributional hypothesis**, according to which, in summary, linguistic items with similar distributions have similar meanings.
This theory has roots in several studies starting in the 1950s in the field of linguistic. The groundbreaking idea that the meaning of a word can be represented as a point in space is due to Osgood et al. ((1957)). In their revolutionary work, *The Measurement of Meaning*, they notice that it is possible to retrieve a 3-dimensional vector from a word, by assigning it ratings on three different scales, decided by the authors.

On the other hand, other scholars in these years help to investigate a new research area, that is the **Distributional Semantics**. The key concept behind this new branch of linguistic is the idea that it is possible to study linguistic elements starting from their distributional properties (distributional hypothesis). Harris ((1954)) gives an other important contribution with his work *Distributional Structure*, published in 1954, where he write:

> Here we will discuss how each language can be described in terms of a distributional structure, i.e. in terms of the occurrence of parts (ultimately sounds) relative to other parts, and how this description is complete without intrusion of other features such as history or meaning.

In the same years, also Firth ((1957)) and Joos ((1950)) support the same view in their works. According to them, the language usage is fundamental to decipher the meaning of the word, which depends on its context. Consequently, words which are used in the same context are likely to be semantically similar.
In order to better understand, let us consider the following incomplete sentences:

1. Can I use your ______________?
2. Sure, the ______________ is in the living room.
3. Shhh, he's on the ______________.
4. Pick up the ______________ and call me.

While many words can appear in some of these contexts, only the words which can appear in all of them are likely to be synonimous. For example, words like *sofa* in 1., 2., and 3., *television* in 1., and 2., *bathroom* only in 1. belong to a similar semantic fields but they are not equivalent. In the same way, a word like *volcano* which cannot appear in none of these sentences belongs to a completely different semantic field.
In contrast, *telephone* and *dog and bone* have exactly the same meaning, and they can occur in all of the four sentences.

Vector semantics Vector semantics derives from the application of linear algebra tools to distributional semantics. Indeed, this model collects distributional information in high-dimensional vectors, using large samples of language data ((Rieger, 1991)). By using different types of algorithms, as we will see in the following sections, each word is matched to a vector accordingly the above-mentioned principle.

1.2 Semantic and syntactic properties of embeddings

1.2.1 Cosine Similarity Measure

In this vector space, beyond the usual euclidean distance, it is possible to define the measure of **cosine similarity**, which highlights both the distributional and the semantic similarity between two vectors. Let $v,\ w \in \mathcal{D}$ be two words and let $\vec{v},\ \vec{w} \in \mathcal{V}$ be the corresponding vectors. The cosine similarity between $\vec{v}$ and $\vec{w}$ is the cosine of the angle between the vectors, and it can be computed as ((Jurafsky and Martin, 2000)):

$$\cos(\vec{v},\ \vec{w}) = \frac{\vec{v} \cdot \vec{w}}{\|\vec{v}\|_2 \|\vec{w}\|_2}$$

Obviously, $\cos(\vec{v},\ \vec{w}) \in [-1, 1]$. When two words are similar, the cosine is larger, which implies that the angle between them is smaller. For example, let us consider the pre-trained embedding given by 50-dimensional GloVe word vectors ((Atienza, 2018)). We have that:

$$cos(\overrightarrow{father},\ \overrightarrow{mother}) = 0.890903844289$$

$$cos(\overrightarrow{ball},\ \overrightarrow{crocodile}) = 0.274392462614$$

Indeed, the words *father* and *mother* are semantically more similar than *ball* and *crocodile.*

1.2.2 Analogy

Another semantic property of embeddings worth mentioning is the analogy, which is able to capture relational meanings ((Jurafsky and Martin, 2000)).

Origin of the concept In 1973, Rumelhart and Abrahamson ((1973)) first show that sets of concepts can be related by analogical relationships in their work *A Model for Analogical Reasoning.* These analogies can be solved using the parallelogram model through simple vectorial operations. In more recent times, this model is applied to famous pretrained embeddings (as Word2Vec ((Mikolov et al., 2013a)) or GloVe vectors ((Levy et al., 2015a)) and turned out to be very effective in bringing out relations of different nature among words.

Definition If we consider three words a, b, c we can find the solution x of the analogy $a : b = c : x$ (a *is to* b *as* c *is to* x) by computing $\text{argmin}_x\ d(x,\ \vec{b} - \vec{a} + \vec{c})$, where d is the euclidean distance ((Jurafsky and Martin, 2000)). Given the optimal result d, we can also write $\vec{b} - \vec{a} + \vec{c} \approx \vec{d}$. Let us consider again the pre-trained word embedding given by 50-dimensional GloVe word vectors ((Zhang et al., 2021)). We can observe:

$$\overrightarrow{woman} - \overrightarrow{man} + \overrightarrow{son} \approx \overrightarrow{daughter}$$

The analogy in this case captures the semantic relation of the pair (*man*, *woman*) which is of the type (male, female), and transposes it on new words.

$$\overrightarrow{China} - \overrightarrow{Beijing} + \overrightarrow{Tokyo} \approx \overrightarrow{Japan}$$

Now, the **semantic** relation represented by this operation is of the type (country, capital).

$$\overrightarrow{worst} - \overrightarrow{bad} + \overrightarrow{big} \approx \overrightarrow{biggest}$$

On the other hand, in this last example, it is possible to observe the interpolation of the **syntactic** relation of the type (basic form, superlative).

1.3 Models

The term *embedding* is given to a specific type of vectorial representation, where the vectors are particularly short and dense. Despite this current conformation, the first experiments to construct vectorial representations of words are based on the **one hot encoding**, producing sparse vectors whose length was given by the size of the vocabulary, usually much more than 300 ((Jurafsky and Martin, 2000)).
One of the most successful tools used to build word embeddings is **Word2Vec**, introduced by the groundbreaking work *Efficient Estimation of Word Representations in Vector Space* by Mikolov et al. ((2013b)) in 2013. Word2Vec is a group of models, based on different neural architectures: **CBOW model** and **Skip-Gram model**, which are better explained in the section 1.3. After the release of Word2Vec tool, one of the best more recent embedding model is Stanford's **GloVe** ((Pennington et al., 2014)), which is based on a different learning algorithm.
Finally, it is important to mention that both Word2Vec and GloVe models produce **static embeddings**, namely they do not change depending on the context of the word. More recently, learning dynamic contextual embeddings have been trained, like the popular family of BERT representations ((Jurafsky and Martin, 2000)).

1.3.1 One Hot Encoding

One hot encoding representation consists of binary vectors of the same size of the vocabulary $\mathcal{D}$. Each word corresponds to a vector containing all zero values except the index of the word itself in the vocabulary, which is marked with a 1 ((Med, 2020)). All the vectors are orthogonal and equally dissimilar to each other, which is also one of the limitations of this method.
Now, for instance, let us consider as a vocabulary the following sentence:

I left home last summer and I went to Paris and Rome.

$|\mathcal{D}| = 10$, which means that each vector has 10 components. Assuming the indices of the words in the vocabulary are given by their order of appearance in the sentence, the one hot encoding representation becomes:

I	[1 0 0 0 0 0 0 0 0 0]
left	[0 1 0 0 0 0 0 0 0 0]
home	[0 0 1 0 0 0 0 0 0 0]
last	[0 0 0 1 0 0 0 0 0 0]
summer	[0 0 0 0 1 0 0 0 0 0]
and	[0 0 0 0 0 1 0 0 0 0]
went	[0 0 0 0 0 0 1 0 0 0]
to	[0 0 0 0 0 0 0 1 0 0]
Paris	[0 0 0 0 0 0 0 0 1 0]
Rome	[0 0 0 0 0 0 0 0 0 1]

Pros and cons On the one hand, one hot encoded vectors are highly intuitive and easy to compute. On the other hand, they require a significant amount of memory when the vocabulary is large, and, in addition, this type of word representation loses the inner meaning of the word in a sentence, since, by its nature, it does not consider a word's context. Equivalently, it does not emphasize the semantic relation between words, since, as previously mentioned, every vector is equally to any other ((Gupta, 2022)) ((Shabou, 2020)). In this regard, let us consider the words of the example: *Paris* and *Rome*, which are both names of european capitals, should result more similar than *Paris* and *summer*, which are respectively a city and a season.

1.3.2 CBOW Model

Creating word representations can be considered as a learning task, as shown in the paper of Mikolov et al. ((2013b)), which presents two learning model architectures. The first one is the **Continuous Bag-of-Words** (or CBOW) model, which tries to predict the target word based on the surrounding words, named **context words** ((Madhukar, 2020)).

Representation of center and context words Given a text corpus, let us consider a **sliding window** of size h, moving one word at time. At each step, the window contains a center word and $h-1$ context words.
Let us provide an example: given $h = 5$ and the following sentence as corpus:

I love giraffes because I love their long necks

if we take *giraffes* as the center word we obtain:

$$\boxed{\textit{I love giraffes because I}}\ \textit{love their long necks}$$

namely, in this case the 4 context words are I, *love*, *because*, and I again. The vocabulary $\mathcal{D}$ of this corpus is made up by I, *love*, *giraffes*, *because*, *their*, *long*, *necks*, whose corresponding one hot encoded 7-dimensional vectors are:

I	[1 0 0 0 0 0 0]
love	[0 1 0 0 0 0 0]
giraffes	[0 0 1 0 0 0 0]
because	[0 0 0 1 0 0 0]
their	[0 0 0 0 1 0 0]
long	[0 0 0 0 0 1 0]
necks	[0 0 0 0 0 0 1]

as shown in the previous section. For each center word, the corresponding context words are embedded in a vector $\vec{x}$ given by the one hot encoded vectors averaged. In the the previous example, considering again *giraffes* as the center word, context words are embedded in

$$\left(\begin{array}{c} I \\ \begin{bmatrix}1\\0\\0\\0\\0\\0\\0\end{bmatrix}\end{array} + \begin{array}{c} love \\ \begin{bmatrix}0\\1\\0\\0\\0\\0\\0\end{bmatrix}\end{array} + \begin{array}{c} because \\ \begin{bmatrix}0\\0\\0\\1\\0\\0\\0\end{bmatrix}\end{array} + \begin{array}{c} I \\ \begin{bmatrix}1\\0\\0\\0\\0\\0\\0\end{bmatrix}\end{array}\right) /4 = \begin{bmatrix}0.5\\0.25\\0\\0.25\\0\\0\\0\end{bmatrix}$$

It is important to notice that the order of the context words is not considered.

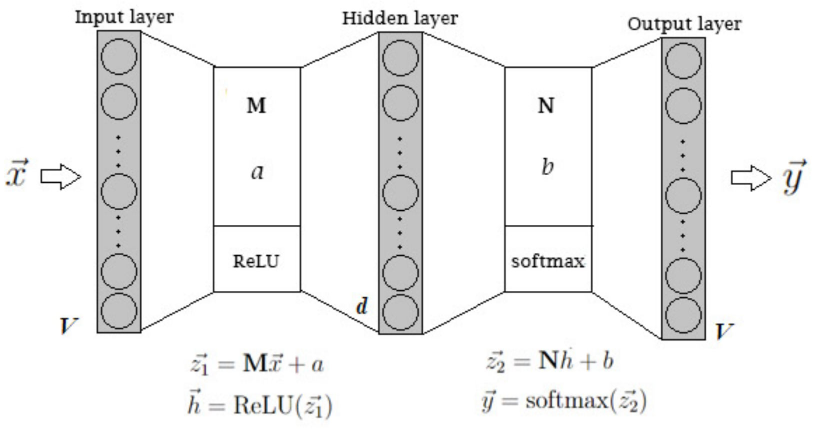

Input layer
Hidden layer
Output layer
M
a
ReLU
N
b
softmax
$\vec{x}$
$\vec{y}$
V
d
V
$\vec{z_1} = \mathbf{M}\vec{x} + a$
$\vec{h} = \text{ReLU}(\vec{z_1})$
$\vec{z_2} = \mathbf{N}\vec{h} + b$
$\vec{y} = \text{softmax}(\vec{z_2})$

Figure 1.1: Schematic representation of the CBOW architecture

The neural network The model consists of a fully connected feedforward neural network, made up by an input layer, an hidden layer and an output layer. The input layer takes as input the context words vector $\vec{x}$, while the output is the predicted center word vector $\underline{\vec{y}}$. The hyperparameters are d, the word embedding size, and $|\mathcal{D}| = N$, the cardinality of the vocabulary. The network is sketched in Figure 1.1. It has ReLU and then softmax activation functions, $\mathbf{M} \in \mathbb{R}^{d \times N}$, $\mathbf{N} \in \mathbb{R}^{N \times d}$, the weights matrices, and a and b the bias values [1]. As loss function, the **Cross Entropy Loss** ((Wikipedia, 2022a)) J is used:

$$J = -\sum_{k=1}^{N} \vec{y_k} \; log(\underline{\vec{y}}_k)$$

considering $\vec{y_k}$ the k-th target word and the corresponding prediction $\underline{\vec{y}}_k$. At the end, the optimal weights matrices **M**, **N** are learned, and the final embedding is given by the raws of the average of **M** and **N** transposed.

1.3.3 Skip-Gram Model

The second architecture can be seen as the reverse of CBOW ((Doshi, 2019)): instead of predicting the current word based on the context, it tries to optimize classification of a word based on another one in the same sentence ((Mikolov et al., 2013b)). In other words, given a word w the learning algorithm tries to predict if a word c is in the context, solving a **classification problem**.

[1]For further information about Neural Networks, see Stanford ((2020))

The classifier In order to define better the classification task, let us consider the following part of sentence and a sliding window of size $h = 5$ ((Jurafsky and Martin, 2000)):

$$\textit{... lemon, a } \boxed{\textit{tablespoon of } \colorbox{lightgray}{\textit{apricot}} \textit{ jam, a}} \textit{ pinch ...}$$

where the target word w is *apricot*.
Given a pair $(w,\ c)$, where c is the candidate context word, the classifier returns the probability that c is a real context word $P(+|w, c)$, which is true if for example $c = jam$ and false if $c = space$. The calculation of $P(+|w, c)$ relies on the intuition that a word is likely to occur near the target if its embedding vector is similar to the target embedding ((Jurafsky and Martin, 2000)). As shown in section 1.2.1, it is possible to quantify similarity between w an c by computing their cosine similarity $cos(\vec{w},\ \vec{c})$ which the normalized dot product between them; in general, $\vec{w}$ and $\vec{c}$ are similar if their dot product $\vec{w} \cdot \vec{c}$ is high. Consequently, the probability $P(+|w, c)$ is defined as:

$$P(+|w, c) = \sigma(\vec{w} \cdot \vec{c}) = \frac{1}{1 + e^{-\vec{w} \cdot \vec{c}}}$$

where indeed σ is the sigmoid function. Since Skip-gram make the assumption that all context words are indipendent, then the probability that a set of h words $c_1, c_2, ...c_h$ are a real context window for w is defined as:

$$P(+|w, c_1, ...c_h) = \prod_{i=1}^{h} \sigma(c_i \cdot w)$$

The algorithm considers as parameters two embeddings for each word, one for the word as a target, and one for the word as context. The two embeddings are gathered respectively in the two matrices $\mathbf{W} \in \mathbb{R}^{d \times N}$ and $\mathbf{C} \in \mathbb{R}^{d \times N}$, whose concatenation gives the matrix θ (where $|\mathcal{D}| = N$ and d is the dimensionality of the embedding).

Learning Skip-Gram with Negative Sampling (SGNS) After taking a corpus of text as input, the skip-gram model begins by assigning a random embedding vector for each of the N vocabulary words. Then, it proceeds to iteratively shift the embedding of each word w to be more similar to the embeddings of words that occur nearby in corpus texts, and less like the embeddings of words that don't occur nearby. The training of this binary classifier is based on the sets of positive and negative samples of the form $(w,\ c)$, which we call respectively

$\mathcal{S}_+ := \{(w,\ c_+)\}$ and $\mathcal{S}_- := \{(w,\ c_-)\}$. Referring to the previous example, having *apricot* as target word, (*apricot, jam*) or (*apricot, tablespoon*) belongs to $\mathcal{S}_+$, while among negative samples belonging to $\mathcal{S}_-$ there could be (*apricot, space*) or (*apricot, dog*). Since the leaning procedure uses more negative samples than the positive ones, it is called **skip-gram with negative sampling** or **SGNS**. The ratio k between the provided pairs $(w,\ c_+)$ and $(w,\ c_+)$ is a hyperparameter of the algorithm: for each positive sample, k negative samples are created according to the weighted unigram frequency of w. Since the goal of this architecture is to maximise similarity between positive $(w,\ c)$ pairs and minimise similarity between negative $(w,\ c)$ pairs, for each word $w \in \mathcal{D}$ the loss function L to minimize is defined as:

$$L(\theta) = -\Big[\log(P(\mathcal{S}_+|w,c)) + \sum_{i=1}^{k} \log(P(\mathcal{S}_-|w,c))\Big] =$$

$$-\Big[\log(\sigma(\vec{w}\cdot\vec{c})) + \sum_{i=1}^{k} \log(\sigma(-\vec{w}\cdot\vec{c}))\Big]$$

This loss function is minimized using stochastic gradient descent[2].
An other hyperparameter is the previous mentioned context window h, which has an heavy influence on the results: a small window size leads to similar embeddings for words with the same semantic role (or the same part-of-speech tag[3]) e.g. *Paris* and *Rome*, while a large window size emphasize the topical relatedness e.g. *pope* and *Rome.* Usually $h \leq 10$.

1.4 Word Alignment

A brief summary The term *Alignment* indicates a mapping function A between two vector spaces corresponding to two different embeddings. Let us consider the embeddings e_1 and e_2 defined as:

$$e_1 : \mathcal{D}_1 \to \mathcal{V}_1 \qquad e_2 : \mathcal{D}_2 \to \mathcal{V}_2$$

where $\mathcal{D}_1$ and $\mathcal{D}_2$ are different vocabularies and $\mathcal{V}_1 \cong \mathbb{R}^{d_1}$ and $\mathcal{V}_2 \cong \mathbb{R}^{d_2}$. Usually, the considered embedding spaces have the same dimensions, hence we denote $d_1 = d_2 = d$. If there exist a correspondence between elements $x \in \mathcal{D}_1$ and $y \in \mathcal{D}_2$, then the alignment between their respective embedding

[2]For a detailed explanation of the skip-gram architecture, see Jurafsky and Martin ((2000))

[3]For further information about part-of-speech or POS tagging, see Engine ((2018))

spaces seeks to find a map A of the following form ((Kalinowski and An, 2020)):

$$A : \mathcal{V}_1 \rightarrow \mathcal{V}_2 \text{ such that } A(e_1(x)) \approx e_2(y)$$

This approach is widely used to build **cross-lingual embedding models**. In this case, referring to the previous notation, $\mathcal{D}_1$ and $\mathcal{D}_2$ are vocabularies in different languages. However, there are example in literature where the alignment is applied to embedding spaces retrieved from data of the same language: an example is given by the project *Histwords* by Hamilton et al. ((2016)), explained in section 3.2.

Cross-lingual transfer The main applications of alignment are cross-lingual embedding models, which are cross-lingual representations of words in a joint embedding space((Ruder et al., 2019)). They can be used to facilitate *cross-lingual transfer*, which consists of modelling on data from one language and then applying it to another relying on shared cross-lingual features, for some NLP tasks.
An example among these cross-lingual tasks is **sentiment analysis**, namely the task of determining the sentiment polarity (e.g. positive and negative) of texts in different languages ((Ruder et al., 2019)). In this context, Mogadala and Rettinger ((2016)) evaluate their embeddings on the multilingual Amazon product review dataset of Prettenhofer and Stein ((2010)).
An other example is obviously **machine translation**, to translate entire texts in other languages. Zou et al. ((2013)) use phrase-based machine translation to evaluate their embeddings[4].

Types of alignment The embedding space alignment can occur in different situations, which influence the types of elements taken as input and returned as output. Our interest lies in **Word** (or Word-to-Word) **Alignment models**, which take as input the embedding of a given word $e_1(x)$ and receive as output the embedding of a semantically or syntactically similar token $e_2(x')$ ((Kalinowski and An, 2020)).
Moreover, two other cases are worth mentioning: **Sentence** (or Sentence-to-Sentence) **Alignment** and **Document Alignment models**. Sentence Alignment models often serve as an entry point to machine translation applications and rely on a parallel corpus of sentences in two languages ((Kalinowski and An, 2020)). Although in the present work we focus on word representations, also sentences can be mapped to vectors of real numbers through *sentence embedding*[5]. Letting s and t be two corresponding sen-

[4]For more example, we recommend to consult Ruder et al. ((2019))

[5]For further information, see Perone et al. ((2018))

tences, and f_1 and f_2 their embeddings, the alignment A is defined in the same way, respecting the property $A(f_1(s)) \approx f_2(t)$. Finally, Document Alignment Models require documents in different languages that are translations of each other ((Ruder et al., 2019)), which is very rare.

1.4.1 Word Alignment Models

Mikolov et al. ((2013c)) first observe that word embedding models trained on different corpora (in their specific case, corpora of distinct languages) exhibite similar geometric patterns and behaviors. This intuition leads to the hypothesis for which word embedding spaces can be transformed from one to another through linear operations. Here we review different **mapping-based approaches**[6] methods for word alignment: starting from embeddings independently trained on different corpora (e_1 and e_2), the goal is to construct the linear map A by finding the transformation matrix $\mathbf{W}$:

$$A : \mathcal{V}_1 \xrightarrow[\vec{x} \mapsto \mathbf{W}\vec{x} \approx \vec{y}]{} \mathcal{V}_2$$

where $\vec{x} = e_1(x) \in \mathcal{D}_1$ and $\vec{y} = e_2(y) \in \mathcal{D}_2$ are representations of **corresponding words** x and y, and $\mathbf{W} \in \mathbb{R}^{d \times d}$. This means that if the alignment is used to build cross-lingual embedding models x and y are each the translation of the other, while if the alignment acts on embedding spaces of the same language, x and y are simply the same word with different representations. Moreover, since A is a bijection, there are some assumptions we can do on $\mathcal{D}_1$ and $\mathcal{D}_2$: $|\mathcal{D}_1| = |\mathcal{D}_2|$ and $\forall x \in |\mathcal{D}_1| \; \exists \; y \in \mathcal{D}_2$ such that y corresponds to x.

Regression methods Regression methods map the source embedding to the target embedding by maximizing their similarity. As shown in *Exploiting Similarities among Languages for Machine Translation* by Mikolov et al. ((2013c)), using the n most frequent words $x_1, ..., x_n \in \mathcal{D}_1$ and the corresponding $y_1, ..., y_n \in \mathcal{D}_2$ as seed words, it is possible to learn a transformation matrix $\mathbf{W}$ using stochastic gradient descent by minimising the squared Euclidean distance (mean squared error, MSE) between $\vec{x}_i$ and $\vec{y}_i$ ((Ruder et al., 2019)). Namely, the function to minimize in this case to find $\mathbf{W} \in \mathbb{R}^{d \times d}$ is defined as:

$$\Omega_{REG}(\mathbf{R}) = \sum_{i=1}^{n} ||\mathbf{R}\vec{x}_i - \vec{y}_i||^2 \text{ with } \mathbf{R} \in \mathbb{R}^{d \times d}$$

[6]Although mapping-based approaches are the most popular, it is important to precise that there exist other approaches, like **pseudo-multi-lingual corpora-based approaches** and **joint-methods** which are well explained in Ruder et al. ((2019)).

which in matrix form becomes:

$$\Omega_{REG}(\mathbf{R}) = \sum_{i=1}^{n} ||\mathbf{RX} - \mathbf{Y}||_F^2 \text{ with } \mathbf{R} \in \mathbb{R}^{d \times d}$$

$\mathbf{X}$, $\mathbf{Y} \in \mathbb{R}^{d \times n}$ are the embedding matrices of the seed words, whose columns of the same index $\mathbf{X}^i$ and $\mathbf{Y}^i$ with $i = 1...n$ are the embedding vectors of two corresponding words $\vec{x}_i$ and $\vec{y}_i$. Finally, it holds:

$$\mathbf{W} = \arg\min_{\mathbf{R} \in \mathbb{R}^{d \times d}} \Omega_{REG}(\mathbf{R})$$

The basic approach of Mikolov et al. ((2013c)) is later used by other researchers who incorporated ℓ_2 regularization ((Ruder et al., 2019)). Sometimes, when $\mathbf{X}$ and $\mathbf{Y}$ are not given, the minimization of $\Omega_{REG} := \Omega_{REG}(\mathbf{R}, \mathbf{X}, \mathbf{Y})$ can be incorporated with the usual learning models used to compute the embeddings[7].

Orthogonal methods This kind of methods is one of the most popular improvement of the previous class of models, by adding the orthogonal constrains, i.e. $\mathbf{W}^T\mathbf{W} = \mathbf{I}$ ((Ruder et al., 2019)). This is equivalent to the **orthogonal Procrustes Problem**, given by:

$$\mathbf{W} = \arg\min_{\mathbf{R} \in \mathbb{R}^{d \times d}} ||\mathbf{RX} - \mathbf{Y}||_F \text{ subject to } \mathbf{R}^T\mathbf{R} = I$$

Schönemann ((1966)) finds out that the exact solution to this problem can be efficiently computed in linear time with respect to the vocabulary size using SVD:

$$\mathbf{W} = \mathbf{VU}^T \quad \text{where} \quad \mathbf{Y}^T\mathbf{X} = \mathbf{U\Sigma V}^T$$

The orthogonal constrains have been motivated in different ways from various authors ((Ruder et al., 2019)): Xing et al. ((2015)) write that these constrains lead to the preservation of length normalization, while Artetxe et al. ((2016)) motivate orthogonality as a means to ensure monolingual invariance and Smith et al. ((2017)) to be self-consistent. However, this class of methods is one of the most used in the current research.

Canonical methods This class of methods is based on the idea of mapping the embeddings in a shared space using **Canonical Correlation Analysis** (CCA). Faruqui and Dyer ((2013)) are the first to apply CCA to project

[7]See the section 1.3.

words from two languages into a shared embedding space. The innovative aspect of this model is that, instead of finding only one transformation matrix $\mathbf{W}$ which projects the source embedding space $\mathcal{V}_1$ into the target embedding space $\mathcal{V}_2$, it is able to learn two transformation matrices $\mathbf{W}^1$ and $\mathbf{W}^2$ projecting respectively the embedding spaces $\mathcal{V}_1$ and $\mathcal{V}_2$ into a new joint space. Relying on the same notation as before, let $e_1(x) = \vec{x}$ and $e_2(y) = \vec{y}$ be the representations of two corresponding words x and y, and let $\mathbf{R}^1, \mathbf{R}^2 \in \mathbb{R}^{d\times d}$ be the possible transformation matrices candidates. The correlation between the projections $\mathbf{R}^1\vec{x}$ and $\mathbf{R}^2\vec{y}$ can be defined as:

$$\rho(\mathbf{R}^1\vec{x}, \mathbf{R}^2\vec{y}) = \frac{\text{cov}(\mathbf{R}^1\vec{x}, \mathbf{R}^2\vec{y})}{\sqrt{\text{var}(\mathbf{R}^1\vec{x})\text{var}(\mathbf{R}^2\vec{y})}}$$

where $\text{cov}(\cdot,\cdot)$ is the covariance and $\text{var}(\cdot)$ is the variance ((Ruder et al., 2019)). Then, the aim of CCA methos is to to maximize the function:

$$\Omega_{CCA}(\mathbf{R}^1, \mathbf{R}^2) = -\sum_{i=1}^{n} \rho(\mathbf{R}^1\vec{x}_i, \mathbf{R}^2\vec{y}_i)$$

which means that

$$\mathbf{W}^1, \mathbf{W}^2 = \underset{\mathbf{W}^1,\mathbf{W}^2 \in \mathbb{R}d\times d}{\arg\max} \; \Omega_{CCA}(\mathbf{R}^1, \mathbf{R}^2)$$

where, as before, $\vec{x}_i$ and $\vec{y}_i$ are the columns of $\mathbf{X}$ and $\mathbf{Y}$. Artetxe et al. ((2016)) prove that the canonical method is similar to the orthogonal method with dimension-wise mean centering ((Ruder et al., 2019)).
As for Ω_{REG} the optimization of $\Omega_{CCA} := \Omega_{CCA}(\mathbf{R}^1, \mathbf{R}^2, \mathbf{X}, \mathbf{Y})$ can be used to compute $\mathbf{X}$ and $\mathbf{Y}$ when they are not given.

Margin methods Finally, this last class of methods, introduced by Lazaridou et al. ((2015)) is built on the optimization of a max-margin based ranking loss instead of MSE in order to reduce *hubness*. Hubness is a phenomenon observed in high-dimensional spaces, occurring when some points are the nearest neighbours of many other points ((Radovanovic et al., 2010)). It often affects cross-lingual word embedding models.
Similarly as before, the aim is to compute the unique transformation matrix $\mathbf{W}$, through the optimization of the new function:

$$\Omega_M(\mathbf{R}) = \sum_{i=1}^{n}\sum_{j\neq i}^{k} \max\{0, \gamma - \cos(\mathbf{R}\vec{x}_i, vecy_i) + \cos(\mathbf{R}\vec{x}_i, \vec{y}_j)\}$$

where $\mathbf{R}\in\ \mathbb{R}^{300\times 300}$ is the generic candidate matrix, γ and k are tunable hyperparameters denoting the margin and the number of negative examples, respectively. Again, Ω_M can be used for the computation of the embedding spaces themselves.

Chapter 2

Linear and Non Linear Optimization Problems

2.1 Linear Programming

In this section, we recall some basic concepts about *linear programming*, in order to introduce in the following ones a series of results concerning *nonlinear programming.* For more detailed information about the topic, we recommend to refer to the textbooks of Bertsimas and Tsitsiklis ((1997)) and Wright and Nocedal ((1999)).

2.1.1 Introduction

A ***linear program*** is a constrained optimization problem in which the function to be maximized (or minimized) is linear and the constraints are described by linear equations and/or inequalities:

$$\begin{array}{cc} \text{max (or min)} & c_1x_1 + \cdots + c_nx_n \\ \text{s.t.} & a_{11}x_1 + \cdots + a_{1n}x_n \sim b_1 \\ & \vdots \qquad\qquad \vdots \\ & a_{m1}x_1 + \cdots + a_{mn}x_n \sim b_m \end{array} \tag{2.1}$$

where the symbol $\sim$ can stand for one of the operators $\leq$, $\geq$, $=$, and $a_{ij}, b_i, c_j \in \mathbb{R}$ $(i = 1, ..., m,, \ j = 1, ..., n)$. The set of all the vectors $x \in \mathbb{R}^n$ satisfying all the constrains, namely the *feasible solutions*, is called the *feasible region.* A vector $\bar{x} \in \mathbb{R}^n$ is an ***optimal solution*** of the linear program if $\bar{x}$ is feasible and $c^T\bar{x} \geq c^T x \ \forall x$ in the feasible region, and the corresponding value $c^T\bar{x}$ is called *optimal value.* Let us provide now an important theorem.

Theorem 1. *Given a linear program, one and only one of the following alternatives holds:*

(a) the problem has at least one optimal solution;

(b) the problem is infeasible, i.e., it has no feasible solutions;

(c) the problem is unbounded, i.e., for every $K \in \mathbb{R}$, there exists a feasible solution x such that $c^T x > K$ ($c^T x < K$ for minimization problems).

2.1.2 Standard form

Every linear program of the form (2.1) is equivalent to a linear program in *standard form*, that is, of the type:

$$\begin{array}{ll} \max & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{array} \tag{2.2}$$

where $A \in \mathbb{R}^{m \times n}$, and $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ and x is a vector of variable in $\mathbb{R}^n$

2.2 Nonlinear Programming

In this section we present the notation we use for nonlinear optimization problems, and we provide some results useful to understand the tools we use in the II part.

2.2.1 Introduction

An optimization problem is defined as the minimization or maximization of a real valued function on a specified set. Letting $\mathcal{F}$ be the *feasible set* and $f : \mathcal{F} \to \mathbb{R}$ be the *objective function*, the problem can be represented as following:

$$\min f(x) \quad \text{s.t. } x \in \mathcal{F} \tag{2.3}$$

or

$$\max f(x) \quad \text{s.t. } x \in \mathcal{F} \tag{2.4}$$

Now, we report some useful definition holding for *minimization problems* (2.3). However, it is important to notice that it is always possible to turn a minimization problem into a *maximization problem* (2.4) by changing the sign of f, and viceversa. In particular, it holds:

$$\max_{x \in \mathcal{F}} f(x) = -\min_{x \in \mathcal{F}}(-f(x))$$

In conclusion there is no loss of generality if we study and address only minimization problems or, on the contrary, only maximization problems.

Definition 1. *An optimization problem of the type (2.3) is said to be infeasible if $\mathcal{F} = \emptyset$, that is, if there are no feasible solutions.*

Definition 2. *An optimization problem of the type (2.3) is said to be unbounded (below) if however, if you choose a value $M > 0$ there is a point $x_M \in \mathcal{F}$ such that $f(x_M) < -M$.*

Definition 3. *An optimization problem of the type (2.3) is said to have an optimal solution if there exists a $x^* \in \mathcal{F}$ such that it results $f(x^*) \leq f(x) \forall x \in \mathcal{F}$. The corresponding value $f(x^*)$ is the* ***optimal value****.*

Optimization problems can be classified, according to the feasible set structure $\mathcal{S}$ into:

- *Continuous Optimization Problems* where variables can assume real values ($x \in \mathbb{R}^n$) and, therefore, we have that $\mathcal{F} \subseteq \mathbb{R}^n$;
- *Integer Optimization Problems* whose variables are constrained to be integers ($x \in \mathbb{Z}$) and, therefore, we have that $\mathcal{F} \subseteq \mathbb{Z}^n$;
- *Mixed Integer Problems* where only a subset of variables is constrained to be integer.

For the purpouse of the present thesis, we will focus only on the first class of the problem.

2.2.2 Continuous Optimization Problems

In order to get a better characterization of the solution points for the above problem, we introduce the following definitions.

Definition 4. *A point $x^* \in \mathcal{F}$ is a* ***global minimum*** *for f in $\mathcal{F}$, if $f(x^*) \leq f(x) \; \forall \; x \in \mathcal{F}$.*

Definition 5. *A point $x^* \in \mathcal{F}$ is a* ***strict global minimum*** *for f in $\mathcal{F}$, if $f(x^*) < f(x) \; \forall \; x \in \mathcal{F}, \; x \neq x^*$.*

Definition 6. *A point $x^* \in \mathcal{F}$ is a* ***local minimum*** *for f in $\mathcal{F}$ if there exists a neighborhood $B(x^*; \; \rho)$, with $\rho > 0$ such that $f(x^*) < f(x) \; \forall \; x \in \mathcal{F} \cap B(x^*; \; \rho)$.*

Moreover, we have an *unconstrained problem* if $\mathcal{F} = \mathbb{R}^n$, or a *constrained problem* if $\mathcal{F} \subset \mathbb{R}^n$.

2.2.3 Existence conditions

When dealing with an optimization problem, it is necessary to check its **well-posedness**. Indeed, there are situations in which there is not a minium point in $\mathcal{F}$ for the function f:

- $\mathcal{F} = \emptyset$
- $\mathcal{F} \neq \emptyset$, but f is unbounded from below on $\mathcal{F}$, i.e. $\inf_{x \in \mathcal{F}} f(x) = -\infty$
- $\mathcal{F} \neq \emptyset$, f is bounded from below on $\mathcal{F}$, but there exists no global minimum point for f in $\mathcal{F}$.

Consequently, sufficient conditions for the existence of a global minimum point can be established.

Existence conditions for continuous problems Now, we report a proposition which consists of a fundamental result regarding the existence of a solution for a given continuous optimization problem. It is strongly related to the well known Weierstrass Theorem.

Proposition 1. *Let $\mathcal{F} \subset \mathbb{R}^n$ be a non-empty and compact set. Let f be a continuous function defined on $\mathcal{F}$. Then there exists a global minimum point for f in $\mathcal{F}$.*

The result only applies to the class of problems with compact feasible set. In the case of constrained problems, since $\mathcal{F} \subset \mathbb{R}^n$, if $\mathcal{F}$ is closed and limited, then it is compact.
Otherwise, when the problem is unconstrained ($\mathcal{F} = \mathbb{R}^n$), or if $\mathcal{F}$ is closed but not limited, it is necessary to identify some subset of $\mathcal{F}$ containing the optimal solutions of problem.

Definition 7. *Let $\mathcal{F} \subseteq \mathbb{R}^n$ and $f : \mathcal{F} \to \mathbb{R}$; a level set for f on $\mathcal{F}$ is a non-empty set of the form:*

$$\mathcal{L}(\alpha) := \{x \in \mathcal{F} : f(x) \leq \alpha\}$$

where $\alpha \in \mathbb{R}$

In particular, if $x_0 \in \mathcal{F}$, we denote by $\mathcal{L}_0$ the level set $\mathcal{L}(f(x_0))$ with $\alpha = f(x_0)$ Now, we provide some propositions useful to establish a sufficient condition for the existence of global optima.

Proposition 2. *Let $\mathcal{F} \subseteq \mathbb{R}^n$ and f be a continuous function defined on $\mathcal{F}$. Suppose there exists a level set of f on $\mathcal{F}$ that it is not empty and compact. Then there exists a global minimum point for f in $\mathcal{F}$.*

Proposition 3. *Let $\mathcal{F} \in \mathbb{R}^n$ and let f be a continuous function on $\mathcal{F}$. Then all the level sets $\mathcal{L}(\alpha) := \{x \in \mathcal{F} : f(x) \leq \alpha\}$ of f in $\mathcal{F}$ are compact if and only if the following condition is satisfied:*

- *Let $\{x_k\}$ be a sequence of $x_k \in \mathcal{F}$ such that $\lim_{k\to\infty} ||x_k|| = \infty$, then it follows that*
$$\lim_{k\to\infty} f(x_k) = \infty$$

By combining propositions 2 and 3, we obtain the following sufficient conditions of existence for a minimization problem with $\mathcal{F}$ closed set.

Proposition 4. *Let $\mathcal{F} \subseteq \mathbb{R}^n$ be a closed set and let f be a continuous function on $\mathcal{F}$ and assume that f is coercive on $\mathcal{F}$, that is*

$$\lim_{k\to\infty} f(x_k) = \infty$$

for each sequence $\{x_k\}$, with $x_k \in \mathcal{F}$, such that $\lim_{k\to\infty} ||x_k|| = \infty$. Then there exists a global minimum of f on $\mathcal{F}$.

2.2.4 Convex and concave programming problems

Convex programming problems *Convex programming problems* consist of an interesting class of problem, since it allows to model plenty of real world applications. We first report some useful definitions and then move to the main features of those problems.

Definition 8. *Let us consider a set $C \subseteq \mathbb{R}^n$. C is a **convex** set if, $\forall x, y \in C$ and $\forall \alpha \in [0,\ 1]$, it holds*

$$\alpha x + (1-\alpha)y \ \in C$$

Definition 9. *Let $C \subseteq \mathbb{R}^n$ be a convex set and let f: $C \to \mathbb{R}$. f is **convex** on C if, given any x, $y \in C$ and $\alpha \in [0,\ 1]$, we have that*

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y)$$

*Then, f is **strictly convex** on C if, given any x, $y \in C$, with $x \neq y$ and $\alpha \in (0,\ 1)$ have that*

$$f(\alpha x + (1-\alpha)y) < \alpha f(x) + (1-\alpha) f(y)$$

Definition 10. *Let $C \subseteq \mathbb{R}^n$ be a convex set and let $f\colon C \to \mathbb{R}$. f is* ***concave*** *on C if, given any $x,\ y \in C$ and $\alpha \in [0,\ 1]$, we have that*

$$f(\alpha x + (1-\alpha)y) \geq \alpha f(x) + (1-\alpha)f(y)$$

Then, f is ***strictly concave*** *on C if, given any $x,\ y \in C$, with $x \neq y$ and $\alpha \in (0,\ 1)$ have that*

$$f(\alpha x + (1-\alpha)y) > \alpha f(x) + (1-\alpha)f(y)$$

Now we can formally define a convex programming problem as following.

Definition 11. *A* ***convex programming problem*** *is a minimization problem of the form:*

$$\min f(x) \quad s.t.\ x \in \mathcal{F}$$

where $\mathcal{F}$ is a convex set and f is convex on $\mathcal{F}$, or, equivalently

$$\max f(x) \quad s.t.\ x \in \mathcal{F}$$

where $\mathcal{F}$ is a convex set and f is concave on $\mathcal{F}$.

Convex programming problems have important properties described in the following proposition, which states the equivalence between local and global minima.

Proposition 5. *Let $\mathcal{F} \subseteq \mathbb{R}^n$ be a convex set and f a convex (strictly convex) function on $\mathcal{F}$. Then any (one) minimum point of f on $\mathcal{F}$ is also (the only) global minimum.*

Concave programming problems Concave programming problems represent another class of minimization problems widely used to model real world applications. Let us see their definition.

Definition 12. *A* ***concave programming problem*** *is a minimization problem of the form:*

$$\min f(x) \quad s.t.\ x \in \mathcal{F}$$

where $\mathcal{F}$ is a convex set and f is concave on $\mathcal{F}$, or, equivalently

$$\max f(x) \quad s.t.\ x \in \mathcal{F}$$

where $\mathcal{F}$ is a convex set and f is convex on $\mathcal{F}$.

Concave programming problems are more difficult than convex ones, since those problems usually have a large number of local minimum points which are not global. Despite this, here we report some results which gives some information about their global minima.

Theorem 2. *Let $\mathcal{F} \subseteq \mathbb{R}^n$ be a closed convex set, and let f be concave (and non-constant) on F. Hence, if there is a global minimum of f on $\mathcal{F}$, this lies on the boundary of $\mathcal{F}$.*

Theorem 3. *Let $\mathcal{F} \subseteq \mathbb{R}^n$ be a polyhedron with at least one vertex and let f be a concave function with global minima in $\mathcal{F}$. Then, there exists a global minimum of f in $\mathcal{F}$ that coincides with a vertex of the polyhedron $\mathcal{F}$.*

The latter theorem shows that, when dealing with concave programming problems with a polyhedral feasible set, the search for a global minimum can focus on the set of vertices of the given feasible set.

2.2.5 Linearization of nonlinear problems

Sometimes, it is possible to *linearize* a nonlinear problem, that is, to transform it into a linear one (of the form 2.1). Among all the examples of this useful operation, here we report just the **linearization of absolute value problems**, which is necessary to understand the computation we do in the II part.

Absolute value problems Let us consider a problem of the following form:

$$\min_{x,y} \sum_j c_j|x_j| + \sum_k d_k y_k \qquad \text{s.t. } (x,y) \in C \tag{2.5}$$

where C, the feasible region, is a polyhedron, and that $c_j \geq 0$. Now, we transform the variables in the following way:

$$x = x_j^+ - x_j^-, \;\; x_j^+ \geq 0, \; x_j^- \geq 0$$

Such transformation is not unique, indeed there are two possible cases:

- $x_j \geq 0 \Rightarrow x_j^+ = x_j + \delta = |x_j| + \delta$ and $x_j^- = \delta$
- $x_j < 0 \Rightarrow x_j^+ = \delta$ and $x_j^- = -x_j + \delta = |x_j| + \delta$

with $\delta \geq 0$. When $\delta = 0$, one component must be 0 and the other one is consequently $|x_j|$. Moreover, we notice that:

$$x_j^+ + x_j^- = |x_j| + 2\delta$$

Now, by making a replacement in the objective function of term $|x_j|$ with the sum of x_j^+ and x_j^-, we obtain that for the optimal solution $x_j^+ = 0$ or $x_j^- = 0$ (or, equivalently, $\delta = 0$). As a result, it is possible to rewrite 2.5 as:

$$\min_{x,y} \textstyle\sum_j c_j(x_j^+ + x_j^-) + \sum_k d_k y_k \qquad \text{s.t.} \qquad \begin{array}{r} (x_j^+ - x_j^-, y) \in C \\ x_j^+ \geq 0,\ x_j^- \geq 0\ \ \forall j = 1, ..., n \end{array}$$

where C is the feasible region.
It is possible to obtain a different formulation, by simply rewriting the absolut value as:

$$|x_j| = \max\{x_j, -x_j\}$$

Then by replacing in 2.5, we have:

$$\min_{x,y} \sum_j c_j \max\{x_j, -x_j\} + \sum_k d_k y_k \qquad \text{s.t.}\ \ (x, y) \in C$$

which, by introducing a new variable z_j, can be rewrite as following:

$$\min_{x,y,z} \textstyle\sum_j c_j z_j + \sum_k d_k y_k \qquad \text{s.t.} \quad \begin{array}{r} z_j = \max\{x_j, -x_j\} \\ (x, y) \in C \end{array}$$

Finally, we obtain:

$$\min_{x,y,z} \textstyle\sum_j c_j z_j + \sum_k d_k y_k \qquad \text{s.t.} \quad \begin{array}{c} -z_j \leq x_j \leq z_j \ \ j = 1, ..., n \\ (x, y) \in C \end{array}$$

Linear regression model Let us suppose we want to build a mathematical (linear) model related to a specific real problem of the form:

$$y = a^T x + b$$

where $x \in \mathbb{R}^n$ is the input vector for the model, $y \in \mathbb{R}$ is the output, $a \in \mathbb{R}^n$ an $b \in \mathbb{R}$ are the parameters related to the model. Moreover, we assume to have a finite set T of *input-output* samples, named *training set*:

$$T := \{(x^1, y^1), ..., (x^m, y^l)\}$$

Then, let us define the error between real and model output:

$$E_i = y^i - (a^T x^i + b)$$

Since our goal is to minimize the errors over the training set $E_i,\ i = 1, ..., l$, a classic approach for calculating model parameters is considering the square loss function and then solving the well-known least-square problem:

$$\min_{a,b} \sum_{i=1}^{l} (y^i - a^T x^i - b)^2 \tag{2.6}$$

Now, it is possible to model the problem by relying on the **absolute value formulation**, namely, by rewriting 2.6 as:

$$\min_{a,b} \sum_{i=1}^{l} |y^i - a^T x^i - b|$$

and then, by using the transformation shown in the section 2.2.5, we obtain:

$$\min_{a,b,z} \sum_{i=1}^{l} z_i \qquad \text{s.t.} \;\; |y^i - a^T x^i - b| \le z_i \quad \forall i = 1, ..., l$$

that is

$$\min_{a,b,z} \sum_{i=1}^{l} z_i \quad \text{s.t.} \quad -z_i \le y^i - a^T x^i - b \le z_i \quad \forall i = 1, ..., l$$

This is the well-known **Least Absolute Deviation** (LAD) model (also known as least absolute residual, least absolute error or least absolute value model)[8].
ℓ_1-norm is a good choice for two reasons: first, the model we get is very easy to solve (since it is equivalent to a linear programming problem); second, ℓ_1-norm is less sensitive to outliers.

2.3 Frank-Wolfe method

The Frank-Wolfe method (aka conditional gradient method or reduced gradient method) is an optimization algorithm originally proposed by Frank and Wolfe ((1956)) to solve quadratic programming problems with linear constraints. In this section we describe the classical method and its properties, in preparation to the II part, where we present a slightly different version used in our experiments.

2.3.1 Description of the algorithm

Let us consider a problem of the form

$$\min f(x) \qquad \text{s.t.} \qquad x \in C$$

[8]For further information, see ((LAD, 2008))

where $f : \mathbb{R}^n \to \mathbb{R}^n$ is a continuously differentiable function and $C \subseteq \mathbb{R}^n$ is a convex compact set.
We start with a feasible solution and, at each iteration, we define a descent direction in the current iterate x_k by solving the problem:

$$\min_{x \in C} \nabla f(x_k)^T (x - x_k)$$

We notice that this is equivalent to minimize the linear approximation of f in x_k:

$$\min_{x \in C} f(x_k) + \nabla f(x_k)^T (x - x_k) \tag{2.7}$$

By the proposition 1, by compactness of C, we can deduce that there exists a solution $\hat{x}_k \in C$ for the linearized problem. Now, let us see a useful proposition:

Proposition 6. *Let $x^* \in C$ be local minimum for problem*

$$\min f(x) \qquad s.t. \qquad x \in C$$

with $C \subseteq \mathbb{R}^n$ convex and $f \in C^1(\mathbb{R}^n)$. Then

$$\nabla f(x^*)^T (x - x^*) \geq 0 \ \forall x \in C$$

which leads to the **first order optimality condition**:

Proposition 7. *Let $C \subseteq \mathbb{R}^n$ be a convex set and $f \in C^1(\mathbb{R}^n)$ be a convex function. $x^* \in C$ is a global solution of the following problem:*

$$\min f(x) \qquad s.t. \qquad x \in C$$

if and only if

$$\nabla f(x^*)^T (x - x^*) \geq 0 \ \ \forall x \in C$$

Returning to 2.7, we have two cases. If $\nabla f(x_k)^T (\hat{x}_k - x_k) = 0$ then we have

$$0 = \nabla f(x_k)^T (\hat{x}_k - x_k) \leq \nabla f(x_k)^T (x - x_k) \quad \forall x \in C$$

and x_k satisfies first order optimality conditions (proposition 7). On the contrary, if $\nabla f(x_k)^T (\hat{x}_k - x_k) < 0$ we have a new descent direction in x_k given by $d_k = \hat{x}_k - x_k$. Thus we can have a new iterate given by $x_{k+1} = x_k + \alpha_k d_k$ with $\alpha_k \in (0, 1]$ calculated by means of a line search.
Finally, we report the pseudocode:

```
1: Choose a point $x_1 \in C$
```

```
2: For k=1,2,...
3:     Set $\hat{x}_k = \arg\min_{x \in C} \nabla f(x_k)^T (x - x_k)$
4:     If $\hat{x}_k$ satisfies some specific condition, then STOP
5:     Set $x_{k+1} = x_k + \alpha_k(\hat{x}_k - x_k)$, with $\alpha \in (0, 1]$ suitably chosen stepsize
6: End for
```

2.3.2 Unit stepsize

There are several rules for choosing the stepsize α_k depending on the structure of the problem. Here we mention only a particular type of **fixed stepsize**, namely a stepsize which remains constant throughout iterations. In other word:

$$\alpha_k = s, \quad \text{with } s > 0, \;\; k = 0, 1...$$

More precisely, in the present thesis we consider the **unit stepsize**[9], that is, $\alpha_k = 1 \;\; k = 0, 1....$ The following result concerns the minimization of a concave function over a polytope, by applying a simplified version of the Frank-Wolfe algorithm with unit stepsize.

Proposition 8. *Let us consider the problem*

$$\min f(x) \qquad s.t. \qquad x \in C$$

with $f \in C^1(\mathbb{R}^n)$ concave function lower bounded on C, and $C \subseteq \mathbb{R}^n$ polyhedron. The Frank-Wolfe algorithm with unit stepsize converges to a stationary point in a finite number of steps.

In other words, thanks to the latter proposition 8, we are sure to find a solution in a finite number of steps by applying such algorithm in that situation.

[9]For an in-depth analysis of such variant of the Frank-Wolfe algorithm, and in general to better understand versatility and applicability in a wide range of contexts of the method, we recommend the paper by Rinaldi et al. ((2021)).

Chapter 3

Cultural Semantic Conditioning: from Idelogical Biases to Language Evolution

Word2Vec pretrained embeddings[10] are freely available online ((Wikipedia2Vec, 2020)). However, the usage of pretrained embeddings requires caution, since these representations can hide and propagate cultural peculiarities inherited by the large corpora of text on which they were trained ((Petreski and Hashim, 2022)). In this chapter we will go through two declinations of the same phenomenon, which we indicate with the term **cultural semantic conditioning**. Such phenomenon affects words embeddings, showing the intrinsic link between data and vectorial representations, whose inner geometries depend on cultural and ideological differences among corpora. Several studies prove that embeddings can reflect the cultural semantic conditioning due to particular prejudice of different types, such as gender bias, racial bias, homophobia and discrimination (as shown in section 3.1) or to the general evolution of a language troughout history (which consists of the phenomenon of *historical semantic change* explained in section 3.2).

3.1 Societal Biases in Word Embeddings

According to the *Oxford English Dictionary* in general the term *bias* means: *Tendency to favour or dislike a person or thing, especially as a result of a preconceived opinion; partiality, prejudice* ((Dictionary, 2021)). In the context of AI fairness, this term occurs very often but it is not defined in a unique and precise way. By various AI researchers, it is described as *skew that result in*

[10]Word2Vec architectures are described in section 1.3

undesirable impacts ((Crawford, 2017)) or more precisely *computer systems that systematically and unfairly discriminate against certain individuals or groups of individuals in favor of others* ((Friedman and Nissenbau, 1996)). As, Blodgett et al. ((2020)) write in *Language (Technology) is Power: A Critical Survey of "Bias" in NLP*:

> Large body of work analyzing "bias" in natural language processing (NLP) systems has emerged in recent years, including work on "bias" in embedding spaces (...). Although these papers have laid vital groundwork by illustrating some of the ways that NLP systems can be harmful, the majority of them fail to engage critically with what constitutes "bias" in the first place.

In the present work we try to provide a general definition, focusing on the context of word embeddings, fully aware that past attempts are criticized for their inconsistency.
In *Bias in Computer Systems* Friedman and Nissenbau ((1996)) propose an interesting framework for analyzing algorithmic biases, splitting them into *preexisting*, *technical*, and *emergent*. Such subdivision is based on the phases of the learning system in which the bias arises: while the preexisting bias is related exclusively on the input data on which the algorithm is trained, technical bias is caused by technical constraints or technical considerations and emergent bias arises in a context of use with real users. Althought word embeddings might face all three types of bias ((Papakyriakopoulos et al., 2020)), we focus on the **preexisting** bias. The expression indicates the phenomenon in which AI systems embody biases that exist independently, exemplifying them. It has roots in social institutions, practices, and attitudes ((Friedman and Nissenbau, 1996)). Moreover, we adopt their expression *societal* bias[11], by which the authors mean *bias that originates from society at large* to denote different types of bias. Certainly, this type of bias is strongly related to social discrimination, namely, the discrimination emerging from members of one social group towards members of another.
In the following section, we list various well-known types of biases, by reviewing different NLP research works presenting various strategies for analyzing and detecting them.

3.1.1 Different types of societal biases

Gender bias One of the most studied type of societal bias affecting NLP algorithms is gender bias, due to implicit *sexism* permeating the society

[11]The expression is adopted also by other authors in more recent times, like Sheng et al. ((2021))

Extreme *she* occupations

1. homemaker	2. nurse	3. receptionist
4. librarian	5. socialite	6. hairdresser
7. nanny	8. bookkeeper	9. stylist
10. housekeeper	11. interior designer	12. guidance counselor

Extreme *he* occupations

1. maestro	2. skipper	3. protege
4. philosopher	5. captain	6. architect
7. financier	8. warrior	9. broadcaster
10. magician	11. figher pilot	12. boss

Figure 3.1: A table from Bolukbasi et al. ((2016)) showing examples of automatically generated analogies for the pair $she - he$

.

which is often reflected in the input data. Such phenomenon might emerge in different ways: many studies provide evidence for gender bias in the association of the pair *male/female* with a semantic domain (e.g. *work/family*) with sets of traits or with specific occupations ((Caliskan et al., 2022)).
In *Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings* of Bolukbasi et al. ((2016)) the authors adopt two approaches to detect these biased associations, focusing mainly on the embedding *w2vNEWS*[12]. First, they highlight **occupational stereotypes** by measuring the distances among given occupational terms and the words *she* and *he* (some results are shown in the figure 3.1). Second, they investigate **analogies exhibiting stereotypes** by automatically generating pairs of words (x, y) which satisfy the relation

$$he \ : \ she = x : y \quad \text{or} \quad \vec{she} - \vec{he} \approx \vec{y} - \vec{x}$$

Taking $(he, she) = (a, b)$, for all the possible pairs (x, y) it is defined the following score:

$$S_{a,b}(x, y) = \begin{cases} \cos \vec{a} - \vec{b}, \vec{x} - \vec{y} & \text{if } ||\vec{x} - \vec{y}|| \leq \delta \\ 0 & \text{otherwise} \end{cases}$$

where δ is a threshold for similarity. The best pairs (x, y) are the one with the largest score $S_{a,b}(x, y)$, and some of them are shown in the figure 3.2:

[12]w2vNEWS consists of the word2vec embedding trained on a corpus of Google News texts consisting of 3 million English words and terms into 300 dimensions (for further information see the section 1.3

Gender stereotype *she-he* analogies.

sewing-carpentry	register-nurse-physician	housewife-shopkeeper
nurse-surgeon	interior designer-architect	softball-baseball
blond-burly	feminism-conservatism	cosmetics-pharmaceuticals
giggle-chuckle	vocalist-guitarist	petite-lanky
sassy-snappy	diva-superstar	charming-affable
volleyball-football	cupcakes-pizzas	hairdresser-barber

Gender appropriate *she-he* analogies.

queen-king	sister-brother	mother-father
waitress-waiter	ovarian cancer-prostate cancer	convent-monastery

Figure 3.2: A table from Bolukbasi et al. ((2016)) showing the list of the occupations that are closest to *she* and to *he* in the w2vNEWS embedding
.

many of them have been judged *stereotype* and others *appropriate* by crowdsourcers[13].

Finally, the authors present a *debiasing algorithm* which tries to correct the components of biased words by finding a *gender subspace*, given by two binary extremes among an axesrelying on pairs of *gender specific words.*
However, gender biases in text can extend beyond such semantic associations among words. Caliskan et al. ((2022)) in *Gender Bias in Word Embeddings: A Comprehensive Analysis of Frequency, Syntax, and Semantics* investigate deeply how gender biases permeate word embeddings by studying the frequency, the *part-of-speech*, the conceptual clusters and the emotional characterization[14] of words associated with each gender.
Finally, we are fully aware that gender is a complex spectrum, instead of a simple polarity. We rely anyway on the polar classification, due to the limitations inherent in data which make a comprehensive group representation difficult.

Racial bias Another important type of social discrimination which might emerge from word embeddings is the one towards specific ethnic groups.

[13]The study of Bolukbasi et al. ((2016)) relies also on human experiments, which were performed on the Amazon Mechanical Turk crowdsourcing platform. Only U.S.-based workers were selected, in order to maintain homogeneity and reproducibility with crowdsourcing to the extent possible.

[14]With *emotional characterization* we indicate the scores given to words in the three components of emotion: valence (the pleasantness of a stimulus), arousal (the intensity of emotion provoked by a stimulus), and dominance (the degree of control exerted by a stimulus). This information is widely used by researchers working on text-based sentiment analysis. Warriner et al. ((2013))

Caliskan et al. ((2017)) show in *Semantics derived automatically from language corpora contain human-like biases* that vectors representing African American names are closer to negative words than positive words, a trend which does not repeat among European American names. In order to define such associations, the authors use analogy as well.
Similarly, Garg et al. ((2018)) find occupational stereotypes related to embeddings of proper names depending on the ethincity. In their experiments, bias detection is achieved by measuring distance among the elements a set of proper names and a set of occupations words[15]. However, other authors highlight the presence of ethnic bias hidden in NLP systems also beyond word-embedding themselves ((Kiritchenko and Mohammad, 2018; Davidson et al., 2019)).

Political bias Gordon et al. ((2020)) prove in their paper that it is possible to apply the same approach of Bolukbasi et al. ((2016)) in order to investigate the political bias contained in word embeddings. Their work models political bias as a binary choice between the Democratic pole and the Republican pole, since they refer to American politics. In order to apply the methodology of Bolukbasi et al. ((2016)), the authors find binary pairs by analyzing the most frequent words of a collection of tweets of politicians of both parties. Finally, they conclude that modelling bias along multiple axes or as a range of points along a single axis could be necessary to make a comprehensive analysis, since the politics, as many other fields, is much more complicated than a binary choice.

Other types of bias This is only a partial overview of the kinds of biases investigated in the relevant literature. Papakyriakopoulos et al. ((2020)) uncover sexual orientation stereotypes in word embeddings trained on Wikipedia and social media[16]. Using the method of ((Bolukbasi et al., 2016)), they bring out the occupational words related to homosexuality, which correspond to stereotypical roles such as *artists* and *hairdressers*. Moreover, beyond occupational context, homosexuality seems to be related with very negative concepts: from violence, prohibition and adultery, to abuse and harassment. Also Sheng et al. ((2021)) mention different types of bias emerging, related to profession, sexuality or other.

[15]For further information about word embeddings and methodologies adopted by the authors, we recommend to read the original papers of Caliskan et al. ((2017)) and Garg et al. ((2018))

[16]Papakyriakopoulos et al. ((2020)) train their own embeddings on data collected from Facebook, Twitter and Wikipedia. For more precise information, we recommend to see the third section of the original paper.

3.1.2 The problem of bias propagation and amplification

Due to their widespread usage, word embeddings have the power to amplify intrinsic stereotypes which hide in language itself. In fact, every day, billions of people interact with interfaces that help them access information and make decisions. Consequently, NLP algorithms built to allow that can influence people's behaviors and perceptions about the world.
In order to understand the entity of the problem, let us see two examples.

Search engine In web search, it has been shown that, when carefully combined with existing approaches, word vectors have the potential to improve web page relevance results. Hence, let us suppose that the search query is *cmu computer science phd student* for a computer science Ph.D. student at Carnegie Mellon University. Between two pages that differ only in the names Mary and Mark, the biased word embedding would influence the search engine to rank John's web page higher than Mary. This means that in general, in certain fields, women could lose visibility, contributing to gender gap.

NLG Tasks Natural Language Generation (NLG) is a set of techniques used to generate human-readable language for different applications, such as virtual assistants, chat bots, automatic translators, summarizers, and creative language composers. Sheng et al. ((2021)) emphasize in their work the importance of understanding how societal biases manifest in NLG applications, which directly interact with many different users (for example, chat bots for health, education, and customer support). As we pointed out before, the authors mention many studies about various types of biases pervading NLG systems. Such biases are due to both the training of NLG algorithms themselves, and to the word embeddings used to represent data.

3.2 Historic Semantic Change

Bloomfield ((1933)) defines *semantic change* in his fundamental work *Language*:

> Innovations which change the lexical meaning rather than the grammatical function of a form, are classed as *change of meaning* or *semantic change.*

Nowadays, it is possible to find many NLP research works investigating the repercussions of such linguistic phenomenon on *diachronic* word embeddings.

The adjective *diachronic* means *of, relating to, or studying the development of a phenomenon through time* ((Dictionary, 2022a)).
In the following sections we will summarize two studies which has been fundamental for the present work, since they have provided tools we used to conduct our analysis[17].

3.2.1 *Histwords*

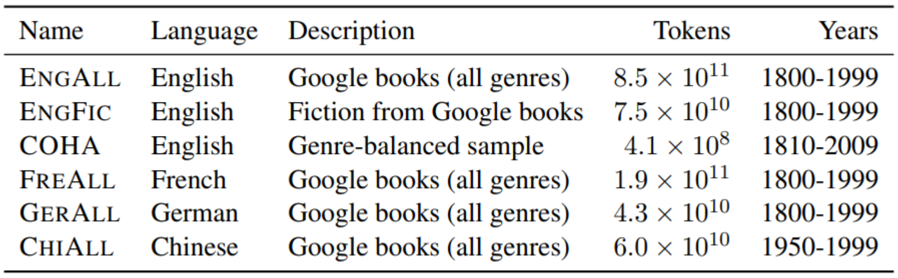
A group of researchers at Stanford University show in their paper *Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change* ((Hamilton et al., 2016)) that it is possible to study the historical semantic change of words by analyzing different embedding spaces. Each of these embedding spaces is obtained by training known learning algorithms over texts written in a specific time period, spanning two centuries, from 1800 to 1990: as a result, the authors provide diachroinic word embeddings, which are publicly available on the GitHub page. These embeddings are then aligned over time and submitted to a procedure to quantify the semantic change. With this purpose, in this study, they develope a new methodology, capable of verifying known semantic shifts and discovering new shifts from the available data.

Datasets The project relies on the historical corpora of Google n-grams ((Lin et al., 2012)) ((GoogleBooks, 2013)) and the Corpus of Historical American English ((Davies, 2012)). On the one hand the Google N-Grams dataset is extremely large, since it includes almost 6% of every book ever published, but on the other hand, it contains many corpus artifacts and noise. On the contrary, the COHA corpus is smaller but cleaner. The 6 corpora, described precisely in the figure 3.3, span 200 years and 4 languages, namely English, German, French, and Chinese.

Embedding Algorithms In order to construct the word embeddings, different state-of-the-art approaches are compared: PPMI, SVD, and SGNS (i.e. Word2Vec) using the work of Levy et al. ((2015b)) as a reference. According to the paper, the latter embedding seems to achieve better performances.

Alignment In order to compute the transformation matrix, the authors consider the Procrustes problem with orthogonal constrains, applied on each pair of diachroinc word embeddings. Let $\mathbf{W}^{(t)} \in \mathbb{R}^{d\times N}$ be the matrix of word embeddings where N is the cardinality of the vocabulary. The alignment

[17]The anaysis is explained in the part II of the thesis.



Name	Language	Description	Tokens	Years
ENGALL	English	Google books (all genres)	8.5×10^{11}	1800-1999
ENGFIC	English	Fiction from Google books	7.5×10^{10}	1800-1999
COHA	English	Genre-balanced sample	4.1×10^{8}	1810-2009
FREALL	French	Google books (all genres)	1.9×10^{11}	1800-1999
GERALL	German	Google books (all genres)	4.3×10^{10}	1800-1999
CHIALL	Chinese	Google books (all genres)	6.0×10^{10}	1950-1999

Figure 3.3: A table from ((Hamilton et al., 2016)) showing the characteristics of the 6 datasets built for the project *Histwords*

.

across time period is realized by optimizing the following problem:

$$\mathbf{R}^{(t)} = \underset{\mathbf{Q}^T\mathbf{Q}=\mathbf{I}}{\arg\min} ||\mathbf{Q}\mathbf{W}^T - \mathbf{W}^{(t+1)}||_F$$

where $\mathbf{R}^{(t)} \in \mathbb{R}^{d\times d}$. The solution is provided by the usual application of SVD provided in section 1.4.

Analysis Finally, the authors present two methods to quantify semantic change over time: (i) by measuring pair-wise word similarities (ii) by quantify an individual overall word's embedding shift, through *Spearman correlation*. In the present thesis we provide a measure similar to (i) for our analysis, which will be explained in part II. Let us see an explanatory example showing some interesting conclusions of this project [18]. The Figure 3.4 shows a representation of the semantic change of the words *gay*, *broadcast*, and *awful* over time, comparing the diachronic Word2Vec embeddings learned at year t. It is possible to see how the context words, which are the grey ones, change depending on the epoch related to the corresponding embedding space.
The word *gay* (**A**) moved away from *cheerful* and *tasteful* and got closer to *homosexual* and *lesbian*, due to its new meaning connected to the sexual orientation semantic field. According the same visualization, *broadcast* (**B**) completely shifted its meaning, from its original sense of sowing seeds, to its current use, connected to contemporary technologies. Finally, *awful* (**C**) changed from meaning *full of awe* to meaning *terrible or appalling.*

[18]For further information, we recommend the original paper ((Hamilton et al., 2016))

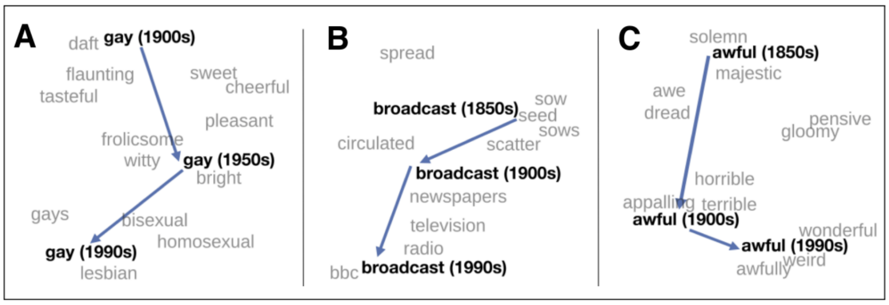


Figure 3.4: A t-SNE visualization of the semantic change of the three words *gay*, *broadcast*, and *awful* over time, taken from Hamilton et al. ((2016))

.

3.2.2 *Exploring Word Evolution*

Another paper is worth mentioning: *Every Word has its History: Interactive Exploration and Visualization of Word Sense Evolution* by Jatowt et al. ((2018)). The authors present an interactive framework that allows users to study the semantic change of words and concepts, made public on the web page Exploring Word Evolution. By using word representations, giving any query word, the online platform provides evolutionary word investigation at different levels: *word analysis, contrastive word pair analysis, multi-word analysis* and *temporal context analysis*.

Datasets The study is based again both on the GoogleBooks ((2013)) 5-gram dataset and on the Corpus of Historical American English (COHA) ((Davies, 2012)). On the website, the user can select one of the two datasets, on which the analysis will be based[19].

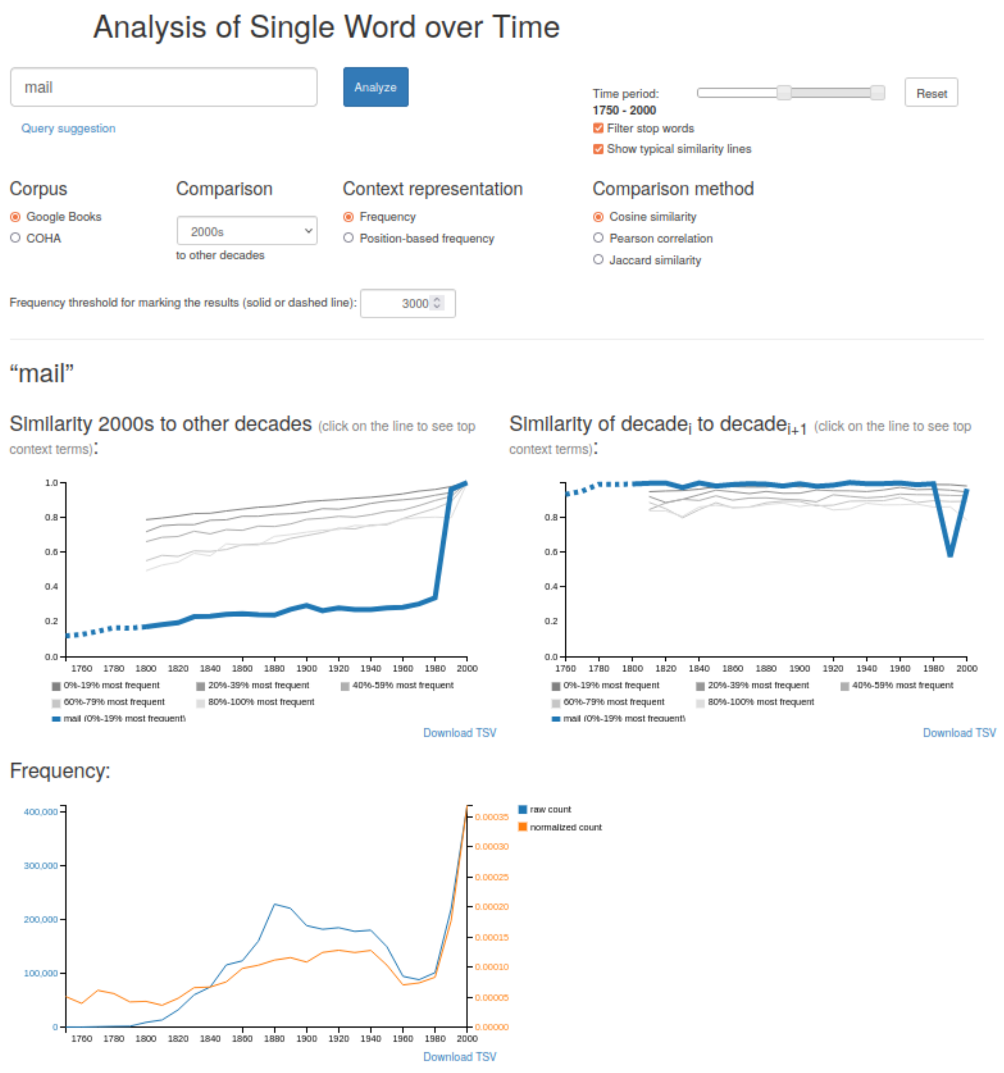
System Description For a given query word w, in a decade d, the system automatically finds the vectorial representation $\vec{w}_d$ [19].
Here we focus on the first type of analysis made by the authors, the *word analysis*, since we use it in our experiments reported in part II. Such test evaluates the degree of a query word's context change across time. The representation $\vec{w}_{\bar{d}}$ at reference decade $\bar{d}$ is compared to any other decade of a fixed time range. In order to do the comparison, the user can chose among the measures *cosine similarity*, *Pearson correlation*, and *Jaccard similarity*. For the purpouse of the present work, the first method is the most suitable one. We refer to the similarity measure between the representations of a word w in the two decades d_1, d_2 provided by the website *Exploring Word*

[19]For further information, we recommend the original paper ((Jatowt et al., 2018)).

Evolution as:

$$\mathbf{s}_E(\vec{w}_{d_1}, \vec{w}_{d_2}) := |\cos(\vec{w}_{d_1}, \vec{w}_{d_2})|$$

Then, also a similarity plot is provided (as shown in the following example), representing the trend of $\mathbf{s}_E(\vec{w}_{\bar{d}}, \vec{w}_{d_i})$ by varying the decade d_i over the selected time period (by default, from 1750 to 2000). By downloading the CSV, the user can visualize specifically all the $\mathbf{s}_E$ values.



Analysis of Single Word over Time

mail

Analyze

Query suggestion

Time period: 1750 - 2000

Reset

☑ Filter stop words

☑ Show typical similarity lines

Corpus: [✓ Google Books] COHA

Comparison: 2000s to other decades

Context representation: [✓ Frequency] Position-based frequency

Comparison method: [✓ Cosine similarity] Pearson correlation, Jaccard similarity

Frequency threshold for marking the results (solid or dashed line): 3000

"mail"

Similarity 2000s to other decades (click on the line to see top context terms):

Similarity of decade$_i$ to decade$_{i+1}$ (click on the line to see top context terms):

Frequency:

Figure 3.5: A screenshot showing the results obtained for the the word *mail* in the online framework Exploring Word Evolution Jatowt et al. ((2018))

.

The screenshot of Figure 3.5 shows the outputs for the the word *mail*. The top left-hand side graph shows the similarity plot between the query's semantics in the reference decade and the ones in each past decade using cosine similarity. The right-hand side plot adds more evidence to the word evolution analysis by outputting similarities between two consecutive decades. It is clearly evident from these charts that the word *mail* has been been subjected to a deep semantic evolution starting from 1980s. Finally, the bottom graph shows both the raw word count and its normalized frequency over time, offering information about the interplay of word popularity and its semantic change.

3.3 Alignment and Cultural Contaminations

Finally, it is important to highlight how cultural peculiarities are not only hidden in the inner structure of word embeddings, but they are also disseminated by alignments[20] to different embedding spaces. Several studies show how this *contamination* can happen in the context of cross-lingual word embeddings, also providing methods to mitigate the consequences on machine translation of such phenomenon.

Girl* and *shōjo Zhang et al. ((2019)) present in their paper a method to increase word translation accuracy of cross-lingual word embedding models, obtained by applying an orthogonal alignment. They notice some difficulties in building cross-lingual mappings from a language to another due to **cultural differences** related to them, and they propose an algorithm (called the *Iterative Normalization*) which tries to correct the monolingual word embeddings before the alignment, in order to make the transformation more effective.
For instance, they focus on the translation between Japanese and English: the usual orthogonal alignment fails in translating the word *shōjo*, which means *girl*. This is due to the fact that in the two monolingual embeddings, the most similar words to *shōjo* and *girl* are different: in the Japanese embedding *shōjo* is closed to *neko*, which means *cat*, while in the English embedding *girl* and *cat* are much more unsimilar. This is due to the fact that in Japanese culture, cats are considered animals belonging to feminine dimension, contrary to what happen in west culture.

Towards more neutral monolingual embeddings Doval et al. ((2018)) state that language-specific phenomena and **corpus specific biases** make

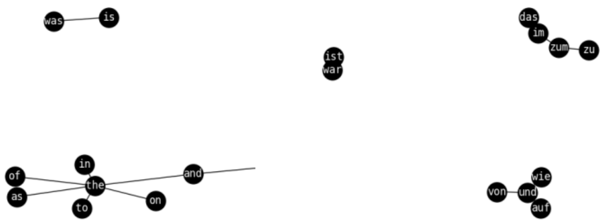
[20]Word alignment is explained in section 1.4

the alignment between two monolingual embeddings more difficult. Consequently, they propose a method to average between the representations obtained from different languages, ending up with more “neutral” monolingual embedding. In particular, they attempt to do it by moving each word vector towards the middle point between its current representation and the representation of its translation.

Different languages, different graphs Søgaard et al. ((2018)) identify the limitations of unsupervised machine translation, finding out that performances are generally worse when monolingual corpora from different domains or different embedding algorithms are used. Regarding this, their critique is based on the idea that:

> Unsupervised approaches to learning crosslingual word embeddings are based on the assumption that monolingual word embedding graphs are approximately isomorphic, that is, after removing a small set of vertices (words)

They use as a reference the state-of-the-art unsupervised model of Conneau et al. ((2017)), which relies on an orthogonal alignment, and attempt to improve it. They firstly investigate the **nearest neighbor graphs** of word embedding spaces in order to conclude that, in general, monolingual word embeddings are far from isomorphic. This holds even if the two languages are closely related, like English and German. In the figure 3.6 it is possible to see how different are the nearest neighbors graph of the embeddings of the 10 most frequent English words and the one of their translation in German. This could be due both to the differences between the synctatic structures, and the different meaning associations between the two languages.



(a) Top 10 most frequent English words (b) German translations of top 10 most frequent English words

Figure 3.6: A representation provided by Søgaard et al. ((2018)) of the nearest neighbor graphs of 10 most frequent words in English Wikipedia and of their their automatic translation in German, by using the method of Conneau et al. ((2017)).

.

Part II

Our Contribution: the Idea and the Experiments

Chapter 4

Contributions

Let us describe the notation we adopt in the whole chapter. It is important to precise that the present work is based on the comparison of two different word embeddings in the same language. Hence, we will refer to them as e_1 and e_2:

$$e_1 : \mathcal{D}_1 \to \mathcal{V}_2 \qquad e_2 : \mathcal{D}_2 \to \mathcal{V}_2$$

where $\mathcal{D}_1$ and $\mathcal{D}_2$ are the vocabularies, and $\mathcal{V}_1$ and $\mathcal{V}_2$ are the corresponding d-dimensional embedding spaces. For the sake of simplicity, we consider $\mathcal{D}_1 = \mathcal{D}_2 = \mathcal{D}$ and we denote $|\mathcal{D}| = N$. Then, given a word $w_i \in \mathcal{D}$, then $e_1(w_i) = \vec{x_i}$ and $e_2(w_i) = \vec{y_i}$. Moreover, all the vectors $\vec{x_i}$ and $\vec{y_i}$ with $i = 1, ..., N$ are the columns of the matrices $\mathbf{X}$, $\mathbf{Y} \in \mathbb{R}^{d \times N}$ respectively.

4.1 The Idea Behind: an Aligment Highlighting Biases

As explained in chapter 3, every word embedding contains cultural peculiarities inherited by data, which can be then propagated through the alignment. The goal of our study is finding a technique based on word alignment[21] able to auomatically highlight the presence of **cultural semantic conditioning** hidden in word representations.

[21]The topic is precisely explained in section 1.4

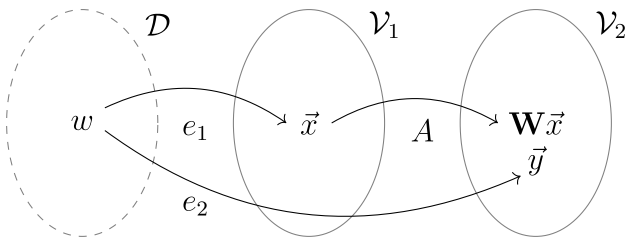


Figure 4.1: Representation of the embedding maps $e_1 : \mathcal{D} \to \mathcal{V}_1$, $e_2 : \mathcal{D} \to \mathcal{V}_2$ and the alignment map $A : \mathcal{V}_1 \to \mathcal{V}_2$ acting in turn on a word w of a vocabulary $\mathcal{D}$.

Let us suppose to have two word embeddings e_1 and e_2 trained on two different corpus of data of the same language reflecting different cultural traits. The goal of a classic alignment $A : \mathcal{V}_1 \to \mathcal{V}_2$ should be minimizing the error among all the words $w \in \mathcal{D}$, namely between the image $A(\vec{x})$ and $\vec{y}$, such that $e_1(w) = \vec{x}$ and $e_2(w) = \vec{y}$, as represented in figure 4.1.
Now, we make an **hypothesis**: we suppose that some words would be changed more by the alignment A due to the semantic cultural conditioning, while other more *stable*, less likely to change semantic meaning depending on societal biases or historic characteristics of the data, would change less. Consequently, in the present thesis, we attempt to change the classic approach to alignment, by starting from a simple idea: instead of looking for a transformation which tries to minimize the error on all $w \in \mathcal{D}$, we want to obtain a map A and a corresponding matrix $\mathbf{W}$ which minimizes the error only on words which *would not drastically change* after applying A. In other words, our ideal alignment would leave the images $A(\vec{x}) = \mathbf{W}\vec{x}$ of representations of words more likely to be subject to cultural semantic conditioning farther from the corresponding $\vec{y}$ than the words more likely to be stable.

4.2 Embedding Alignment: A Nonlinear Programming Approach

In this section we describe the three alignments A_1, A_2, and A_3 used in our experiments: starting from the classic approach of orthogonal alignment A_1, we move then to the computation of two alignment model A_2 and A_3, based on different optimization problems.
In all the cases, the desired alignement A_i is a map of the type:

$$A_i : \mathcal{V}_1 \xrightarrow[\vec{x} \mapsto \mathbf{W}_i \vec{y}]{} \mathcal{V}_2$$

where $\mathcal{V}_1$ and $\mathcal{V}_2$ are d-dimensional embedding spaces, and $i = 1, 2, 3$. As deeper explained in section 1.4, in order to find the map A it is necessary to find a transformation matrix $\mathbf{W}_i \in \mathbb{R}^{d \times d}$.

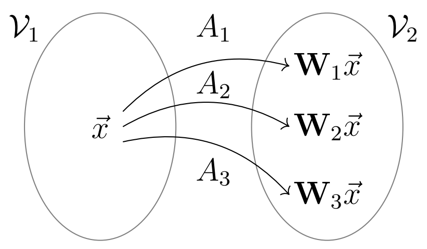


Figure 4.2: Representation of the alignment maps $A_i : \mathcal{V}_1 \to \mathcal{V}_2$, $i = 1, 2, 3$ acting on a word vector $\vec{x}$ of the first embedding space $\mathcal{V}_1$.

4.2.1 A_1: the classic alignment

In order to compute the first alignment matrix $\mathbf{W}_1$ we apply the orthogonal method[22], because of its popularity and its widespread usage, using as a reference *Histwords* by Hamilton et al. ((2016)). The computation of $\mathbf{W}_1$ is based on the resolution of the classic Procrustes problem with orthogonal constrains given by

$$\mathbf{W}_1 = \underset{W \in \mathbb{R}^{d \times d}}{\arg\min} ||\mathbf{WX} - \mathbf{Y}||_F \text{ such that } \mathbf{W}^T\mathbf{W} = \mathbf{I}$$

which can be solved by the application of SVD ((Schönemann, 1966)):

$$\mathbf{W}_1 = \mathbf{UV}^T, \text{ with } \mathbf{U}\Sigma\mathbf{V}^T = SVD(\mathbf{YX}^T)$$

4.2.2 A_2: an alignment optimized by a linear decomposition

In order to obtain the second alignment matrix $\mathbf{W}_2$, we consider the following optimization problem:

$$\mathbf{W}_2 = \underset{W \in \mathbb{R}^{d \times d}}{\arg\min} ||\mathbf{WX} - \mathbf{Y}||_1$$

which coincides to:

$$\mathbf{W}_2 = \underset{W \in \mathbb{R}^{d \times d}}{\arg\min} \sum_{i=1}^{N} ||\mathbf{W}\vec{x_i} - \vec{y_i}||_1$$

[22]The method is explained in the section 1.4

where $\vec{x_i}$ and $\vec{y_i}$ are the i-th columns of the matrices $\mathbf{X}$ and $\mathbf{Y}$. Let $\vec{x_i} := x_i$, $\vec{y_i} := y_i$, let the j-th element of $\vec{y_i}$ be y_i^j, and the j-th row of $\mathbf{W}$ be w^j. Then the formulation above becomes:

$$\mathbf{W}_2 = \arg\min_{W \in \mathbb{R}^{d\times d}} \sum_{i=1}^{N} \sum_{j=1}^{300} |(w^j)^T x_i - y_i^j| = \arg\min_{W \in \mathbb{R}^{d\times d}} \sum_{j=1}^{300} \sum_{i=1}^{N} |(w^j)^T x_i - y_i^j|$$

Consequently, by the nature of the problem, it is possible to solve the latter formulation by combining the solutions of 300 subproblems of the type:

$$\min_{w^j \in \mathbb{R}^d} \sum_{i=1}^{N} |(w^j)^T x_i - y_i^j| \quad \forall\, j = 1...300$$

Each of these non-linear problems behaves like the linear regression model explained in the section 2.2.5, and consequently it can be replaced by the following equivalent linear problem[23]:

$$\min_{w^j \in \mathbb{R}^d,\, z \in \mathbb{R}^N} \sum_{i=1}^{N} z_i \quad s.t. \; - z_i \leq (w^j)^T x_i - y_i^j \leq z_i \quad \forall\, j = 1...300$$

where $z \in \mathbb{R}^N$ is a new variable.
The subdivision into smaller problems is key to the solution of the problem, otherwise the computer would have to deal with an optimization problem with a number of constrains near to 5 millions at once, which would make the computation infeasible.
Through the modeling language AMPL, we apply CPLEX algorithms to the data, finding the optimal $\mathbf{W}_2$ for the problem[24].

4.2.3 A_3: improving the alignment thorugh Frank-Wolfe method

For the computation of the third alignment matrix $\mathbf{W}_3$ we start from the following formulation, based on the 0-norm, considering $\mathbf{W} \in \mathbb{R}^{d\times d}$, $\mathbf{A} \in \mathbb{R}^{d\times N}$, $z \in \mathbb{R}^N$:

$$\mathbf{W}_3 = \arg\min_{\mathbf{W},\mathbf{A},z} ||z||_0 \text{ s.t. } - a_i \leq \mathbf{W}x_i - y_i \leq a_i, \quad e^T a_i = z_i$$

where a_i is the i-th column of $\mathbf{A}$ as usual, and e is a vector of 1s. In our case the 0-norm should lead to a better solution for our purposes, since it

[23]The linearization is realized by using as a reference the method explained in the subsection 2.2.5.

[24]The complete AMPL script can be found in the appendix A.

minimize the error function z by distributing less its components between 0 and the maximal value.
It is possible then to rely on the following approximation of the objective function:

$$||z||_0 \approx \sum_{i=1}^{N}(1 - e^{-\alpha z_i}) = f(z)$$

where α is a fixed positive integer. Hence, we obtain a concave function, given by a finite sum of concave functions, depending only on the variable z. Consequently, although the problem has three variables, which would have to be concatenated in the unique variable x of the pseudocode we provided in the section 2.7, it holds

$$\nabla f(x_k) \cdot x = \nabla f(z_k) \cdot z$$

since $\frac{\partial f}{\partial w_j^l} = \frac{\partial f}{\partial a_i^j} = 0 \ \forall \ j = 1, .., d, \ l = 1, ..., d, \ i = 1, ..., N$. As a result, the minimization problem on which the computation of $\mathbf{W}_3$ relies becomes:

$$\min_{\mathbf{A},\mathbf{W},z} \nabla f(z_k) \cdot z \qquad s.t. \qquad -a_i \leq \mathbf{W}x_i - y_i \leq a_i, \quad e^T a_i = z_i$$

Let us call Ω the feasible set containing all the possible $\mathbf{W} \in \mathbb{R}^{d \times d}$, $\mathbf{A} \in \mathbb{R}^{d \times N}$, $z \in \mathbb{R}^N$ satisfying the constrains above. Now, let us see the pseudocode of the version of the Frank-Wolfe algorithm we use. $z_k \in \mathbb{R}^N$, $g_k := \nabla f(z_k) \in \mathbb{R}^N$ and it are parameters which change value throughout the iterations of the algorithm, while ϵ, used for the cut-off condition, is fixed. At the beginning of the algorithm, $(g_k)_i$ is initialized as $(g_0)_i$

```
 1: $it \leftarrow 0$
 2: $(g_k)_i \leftarrow (g_0)i \ \forall i = i, ...N$
 3: $\mathbf{A}, \mathbf{W}, z \leftarrow \arg\min_{\mathbf{A},\mathbf{W},z} \nabla f(z_k) \cdot z$ such that $\mathbf{A}, \mathbf{W}, z \in \Omega$
 4: $it \leftarrow 1$
 5: **repeat**{
 6:     $(z_k)_i \leftarrow z_i \ \forall i = i, ...N$
 7:     $(g_k)_i \leftarrow \alpha e^{-\alpha (z_k)_i} \ \forall i = i, ...N$
 8:     $\mathbf{A}, \mathbf{W}, z \leftarrow \arg\min_{\mathbf{A},\mathbf{W},z} \nabla f(z_k) \cdot z$ such that $\mathbf{A}, \mathbf{W}, z \in \Omega$
 9:     $it \leftarrow it + 1$ }
10: **until** $\sum_{i=1}^{N} (g_k)_i (z_i - (z_k)_i) \geq -\epsilon$
```

However, the processing of such algorithm would require again the resolution of a problem containing almost 5 millions of constrains at each iteration (in line 3 and then in line 8). It is possible to realize a subdivision of the problem into smaller problems in order to make the computation lighter, by the following substitutions:

$$e^T a_i = \sum_{j=1}^{300} a_i^j = z_i \Rightarrow$$

$$\Rightarrow \nabla f(z_k) \cdot z = \sum_{i=1}^{N} (g_k)_i z_i = \sum_{i=1}^{N} (g_k)_i \sum_{j=1}^{300} a_i^j = \sum_{i=1}^{N} \sum_{j=1}^{300} (g_k)_i \ a_i^j = \sum_{j=1}^{300} \sum_{i=1}^{N} (g_k)_i \ a_i^j$$

Consequently the optimization problem becomes:

$$\min_{\mathbf{A},\mathbf{W}} \sum_{j=1}^{300} \sum_{i=1}^{N} (g_k)_i \ a_i^j \quad s.t. \quad -a_i \leq \mathbf{W}x_i - y_i \leq a_i$$

Since $f(z)$ is a concave function, $(g_k)_i > 0 \ \forall \ i = 1, ..., N$, which means that the previous problem can be solved by combining the solutions of 300 subproblems of the type:

$$\min_{a^j, w^j} \sum_{i=1}^{N} (g_k)_i \ a_i^j \quad \text{s.t.} \quad -a_i^j \leq (w^j)^T x_i - y_i \leq a_i^j \quad \forall \ j = 1, ..., 300$$

where $w^j \in \mathbb{R}^d$ and $a^j \in \mathbb{R}^N$ are the j-th rows of $\mathbf{W}$ and $\mathbf{A}$. Similarly as before, let us call Ω_j the feasible set containing all the possible $w^j \in \mathbb{R}^d$ and $a^j \in \mathbb{R}^N$ respecting the constrains above, for each j. In this case, the pseudocode can be modified in the following way:

```
1: $it \leftarrow 0$
2: $(g_k)_i \leftarrow (g_0)_i \ \forall i = i, ...N$
3: $a^j, w^j \leftarrow \arg\min_{a^j, w^j} \sum_{i=1}^{N} (g_k)_i \ a_i^j$ such that $a^j, w^j \in \Omega_j \ \ \forall \ j = 1...300$
4: $it \leftarrow 1$
5: **repeat**{
6:     $(z_k)_i \leftarrow \sum_{j=1}^{300} a_i^j \ \forall i = 1, ...N$
7:     $(g_k)_i \leftarrow \alpha e^{-\alpha (z_k)_i} \ \forall i = 1, ...N$
8:     $a^j, w^j \leftarrow \arg\min_{a^j, w^j} \sum_{i=1}^{N} (g_k)_i \ a_i^j$ such that $a^j, w^j \in \Omega_j \forall \ j = 1...300$
9:     $it \leftarrow it + 1$ }
10: **until** $\sum_{i=1}^{N} (g_k)_i (\sum_{j=1}^{300} a_i^j - (z_k)_i) \geq -\epsilon$
```

We are certain that the algorithm converges in a finite number of steps by the proposition 8. Indeed, the function f to minimize is concave, and the feasible sets Ω_j are polyhedrons, since they are intersections of linear constrains.

Moreover, it is possible to obtain a further reduction of the dimension of the problem by applying the idea proposed by Rinaldi et al. ((2008)), which is based on the following consideration:

$$\text{if } (z_k)_i = 0 \Rightarrow \sum_{i=1}^{300} a_i^j = 0 \Rightarrow a_i^j = 0 \quad \forall\, j = 1, ..., 300 \Rightarrow$$

$$\Rightarrow\; 0 \leq (w^j)^T x_i - y_i \leq 0 \Rightarrow (w^j)^T x_i - y_i = 0$$

Consequently, let us define the set of indices $I_0 = \{i = 1, ..., N \text{ s.t. } (z_k)_i \neq 0\}$, and the j-th minimization subproblem becomes:

$$\begin{aligned} \min_{a^j, w^j} \quad & \sum_{i \in I_0} (g_k)_i \; a_i^j \\ \text{s.t.} \quad & (w^j)^T x_i - y_i = 0 \;\; \forall\, i \notin I_0 \\ & -a_i^j \leq (w^j)^T x_i - y_i \leq a_i^j \;\; \forall\, i \in I_0 \end{aligned}$$

Let us call $\bar{\Omega}_j$ the space containing all the possible $w^j \in \mathbb{R}^d$ and $a^j \in \mathbb{R}^N$ respecting the new constrains above, for each j. Finally, the pseudocode can be modified by replacing the lines 3 and 8 with:

$a^j, w^j \leftarrow \arg\min_{a^j, w^j} \sum_{i \in I_0} (g_k)_i \; a_i^j$ such that $a^j, w^j \in \bar{\Omega}_j \;\; \forall\, j = 1...300$

Again, through the modeling language AMPL and CPLEX algorithms it is possible to find the optimal solution for $\mathbf{W}_3$[25].

4.3 Measuring the Similarity of Embeddings

4.3.1 Absolute cosine similarity $\mathfrak{s}_i$

One of the fundamental measures for our experiments is $\mathfrak{s}_i$ which takes as argument the transformation matrix $\mathbf{W}_i$. $\mathfrak{s}_i$, applied to the word w, is defined as:

$$\mathfrak{s}_i(w) := |\cos(\vec{y}, \mathbf{W}_i \vec{x})|$$

where $\vec{y} \in \mathcal{V}_1$ and $\vec{x} \in \mathcal{V}_2$ are the representations of w in the two embedding spaces, and $\cos(\cdot, \cdot)$ is the cosine similarity measure provided in section 1.2.1.

[25]The complete AMPL script can be found in the appendix B.

4.3.2 The θ measure

An other important measure for the present project is σ_k, defined as following:

$$\theta_k : \mathcal{V}_1 \times \mathcal{V}_2 \to \mathbb{R}$$

$$\theta_k(\vec{x}, \vec{y}) = \frac{|\{x_1, ..., x_k\} \cap \{y_1, ..., y_k\}|}{k}$$

where $\{x_1, ..., x_k\}$ and $\{y_1, ..., y_k\}$ are the sets of k words corresponding to the nearest k vectors to $\vec{x}$ and $\vec{y}$ respectively in the corresponding embedding spaces $\mathcal{V}_1$, and $\mathcal{V}_2$. Given a word $w \in \mathcal{D}$, and we use θ_k it in two ways:

(i) $\theta_k(e_1(w), e_2(w))$ quantifies how many neighbors have in common the representations of the same word in the two different embedding spaces $\mathcal{V}_1$ and $\mathcal{V}_2$.

(ii) $\theta_k(A(e_1(w)), e_2(w))$ quantifies how many neighbors have in common in the embedding space $\mathcal{V}_2$ the representation $e_2(w) \in \mathcal{V}_2$ of the word w in the second embedding and the aligned image of the representation of the same word w in the first embedding $e_1(w) \in \mathcal{V}_1$.

In the figure 4.3 a diagram representing how θ can act in different embedding spaces is reported: the dashed lines connects the possible arguments of θ, which are $e_1(w) := \vec{x}$, $e_2(w) := \vec{y}$, and $A(e_1(w)) := \mathbf{W}\vec{x}$ (were $\mathbf{W}$ is the matrix associated to the alignment A).

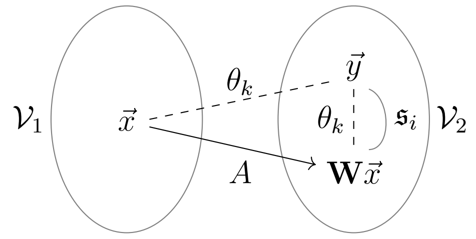


Figure 4.3: Diagram representing how θ_k can act between different embedding spaces.

As shown in the sections 5.2, $\mathfrak{s}_i$ is coherent with θ_k applied as in (ii). Such behaviour is due to embedding models property: indeed, it is easy to evince that if the cosine similarity between two vectors is high, they are likely to have neighbors in common.

The choice of k parameter By varying k among the values 10, 20, 30, 50, we obtain more staggered values as the parameter increases. We observed that higher values of k correspond to heavier computational cost, without achieving a consistently better measure for $k \geq 20$. Consequently, we fix the parameter k to the value 20, and we use the notation $\theta_{20} := \theta$.

Chapter 5

Experiments

This chapter is dedicated to the description of two experiments fundamental for the present work. In the **first experiment**, reported in section 5.1, we propose and validate three test sets, which, according to our hypothesis, should be subject to semantic cultural conditioning in different amounts.
In the **second experiment**, reported in the section 5.2, we compare three different alignments, in order to find a transformation able to highlight cultural peculiarities, as we describe in the section 4.1.

Our focus It is important to precise that due to the lack availability of open source pre-trained word embeddings trained on coeval corpora subject to different cultural influences, the present thesis focuses mainly on one of the declinations of semantic cultural conditioning. Since diachronic word embeddings (i.e. trained on corpora of different epochs) are more easily found online, the **historic semantic change** consists of the core of our research. However, we include also a test set containing words related to societal biases. The embeddings training is a long and computationally expensive procedure, hence we leave the application of our method on coeval word embeddings for future works.

5.1 First Experiment
Validation of test sets

5.1.1 Experimental setup

Pre-trained embeddings We use the pre-trained 300-dimensional Word2Vec historical vectors for English from the GitHub page of the project *Histwords*((Hamilton et al., 2016)), described in the subsection 3.2.1. In particular, we rely on the ones trained on the corpora EngFic, made up by english fiction book, with 7.5×10^{10} tokens. The twenty available embeddings have different vocabulary, as shown in the table 5.2.

Word2Vec embeddings of *Histwords* trained on EngFic	
Decade	Cardinality of the vocabulary
1800	686
1810	1103
1820	1750
1830	2665
1840	3355
1850	4499
1860	4385
1870	4742
1880	6184
1890	8896
1900	9719
1910	7735
1920	10225
1930	9163
1940	8657
1950	11682
1960	13755
1970	13521
1980	19353
1990	24049

Table 5.2: Table illustrating the cardinality of the vocabulary depending on the decade of data of the pre-trained word embeddings of the project *Histwords*.

Selected decades We compare only the representations corresponding to the decades 1890 and 1990, given by embedding maps e_{1890} and e_{1990}:

$$e_{1890} : \mathcal{D}_{1890} \to \mathcal{V}_{1890} \qquad e_{1990} : \mathcal{D}_{1990} \to \mathcal{V}_{1990}$$

Let $\mathcal{D}_{1890}$ and $\mathcal{D}_{1990}$ be the vocabularies related to e_{1890} and e_{1990} respectively: while $|\mathcal{D}_{1890}| = 8896$, and $|\mathcal{D}_{1990}| = 24049$, the cardinality of the intersection $\mathcal{D}_{1890} \cap \mathcal{D}_{1990} := \mathcal{D}$ of the two vocabularies is $|\mathcal{D}| = 8810$.
Then we consider the corresponding 300-dimensional vector spaces $\mathcal{V}_{1890}$ and $\mathcal{V}_{1990}$ containing the diachronic vectorial representations, retrieved by Hamilton et al. ((2016)) through SGNS model. Letting $d = 300$ and $N = |\mathcal{D}| = 8810$, we creat two matrices $\mathbf{X},\mathbf{Y} \in \mathbb{R}^{d \times N}$, whose columns are respectively $\vec{x}_i \in \mathcal{V}_{1890}$ and $\vec{y}_i \in \mathcal{V}_{1990}$, embedding vectors for all the words $w_i \in \mathcal{D}$, $i = 1, .., N$.

Test sets The preparation of the test sets $\mathcal{H}$, $\mathcal{B}$, and $\mathcal{S}$, shown in tables 5.3, 5.4, 5.5, is a fundamental step for the experiments.
$\mathcal{H}$ is the set of words subject to historical semantic change according to literature[26]. $\mathcal{B}$ is the set of words subject to certain types of cultural biases, due to ageism, racism, sexism, or homophobia which are personally annotated or found on the web ((ONGIG, 2020)). $\mathcal{S}$ is the set of words which are likely to be *stable*, namely not subject to biases or semantic change. The idea behind this last list, is that words belonging to the categories of *inaninimate objects* and *common animals and natural elements* are less involved in cultural change over time.

$\mathcal{H}$
fun, fond, terrific, tremendous, awe, grin, smart, egregious, sad, smug, facetious, bully, gay, fatal, awful, nice, broadcast, monitor, record, guy, call, awesome, terrible, terrific, naive, demagogue, guy, mouse, queer, nigger, jaw, kill, astound, knave, knight, recording, bitch, tape, checking, diet, sex, plastic, transmitted, peck, honey, hug, windows, bush, apple, sink, click, handle, instant, twilight, rays, streaming, ray, delivery, combo, candy, rally, snap, mystery, stats, sandy, shades, god, propaganda, atomic, toilet, halloween, king

Table 5.3: Test set of the words subject to historical semantic change

[26]With this regard, the reference texts are ((Hamilton et al., 2016)), ((Floss, 2015)), ((Wikipedia, 2022b)), ((Bloomfield, 1933)), ((Kulkarni et al., 2015)) ((Jatowt and Duh, 2014)), ((Wijaya and Yeniterzi, 2011)).

$\mathcal{B}$
chink, colored, indian, african, foreign, lesbian, gypsy, elderly, handicapped, homos, homosexual, alien, master, slave, retarded, tranny, tribe, girl, boy, man, woman, housekeeper

Table 5.4: Test set of the words subject to societal bias

$\mathcal{S}$
house, tree, table, lamp, book, shoe, mirror, box, fork, chair, telephone, bottle, stove, engine, wallet, boat, pencil, box, cup, plate, paper, stereo, leaf, stick, cloud, shampoo, hat, painting, clothes, watch, window, key, pillow, water, fire, book, door, street, path, bird, horse, cat, dog, fox, fish, school, paper, fountain, cage, ink, pen, bone, forniture, dictionary, umbrella, scissor, hammer, rubbish

Table 5.5: Test set of the words likely to be *stable*.

In $\mathcal{D}$ 25 words of $\mathcal{H}$, 9 words of $\mathcal{B}$, 6 words of $\mathcal{S}$ are missing, so they are not considered for the next analysis.

5.1.2 Results of the experiment

Firstly, we calculate $\theta(\vec{x}, \vec{y})$[27] of all the words $w \in \mathcal{D}$, with $\vec{x} \in \mathcal{V}_{1890}$ and $\vec{y} \in \mathcal{V}_{1990}$ representations of w. The trend of θ values sorted in ascending order, computed for each word in the common vocabulary, is shown in the graph 5.1. It is clear that the representations $\vec{x} \in \mathcal{V}_{1890}$ and $\vec{y} \in \mathcal{V}_{1990}$ of the same word $w \in \mathcal{D}$ change in different ways according to θ. We recall that the higher is θ, the higher the number of common neighbours between two words, which corresponds to an higher semantic similarity. Indeed, θ takes all its possible values throughout the vocabulary, which are discrete by its very nature. Consequently, our hypothesis is confirmed by this behaviour: some words seems to change their meaning more than others, depending on their common neighbors, between the two diachronic embedding spaces.

We validate the **test sets** shown in the tables 5.3, 5.4, 5.5 by computing the averages of $\theta(\vec{x}, \vec{y})$ and the corresponding variances $\sigma_{\theta(\vec{x},\vec{y})}$, for w taken in $\mathcal{H}$, $\mathcal{B}$, and $\mathcal{S}$.

[27] θ is described in the section 4.3

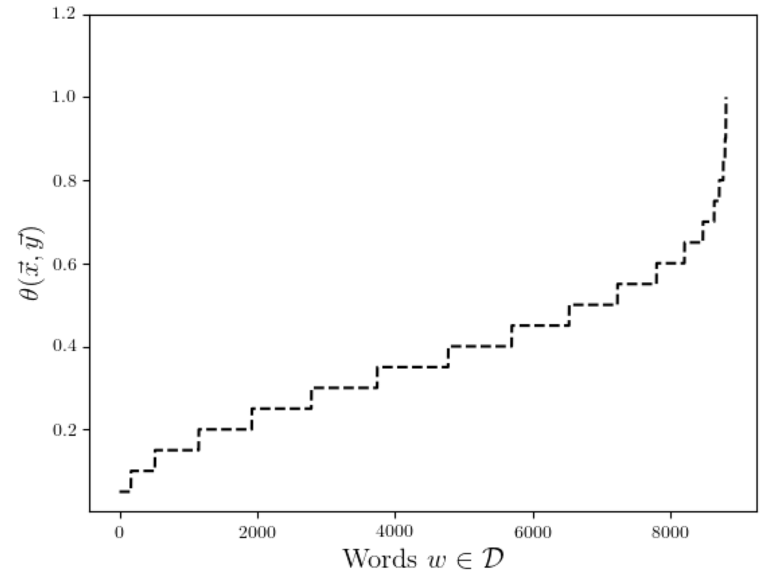


Figure 5.1: Graph representing the values of $\theta(\vec{x},\vec{y})$ reported on the y-axis, for all $w \in \mathcal{D}$ whose indices are reported on x-axis, sorted in ascending order according to the corresponding θ value

Set	$\mathcal{H}$	$\mathcal{B}$	$\mathcal{S}$
$\theta(\vec{x},\vec{y}) \pm \sigma_{\theta(\vec{x},\vec{y})}$	0.30094± 0.04532	0.38437 ± 0.05772	0.497115 ± 0.023789

Table 5.6: Averages and corresponding variances of $\theta(\vec{x},\vec{y})$ values for the test sets

In the table 5.6 the results of validation are shown, confirming our choice of words, especially regarding $\mathcal{H}$. Indeed, the averages of $\theta(\vec{x},\vec{y})$ of the elements of $\mathcal{H}$ and $\mathcal{B}$ are clearly lower than the mean value of $\mathcal{S}$. However, the mean θ of $\mathcal{H}$ is 0.8 lower than $\mathcal{B}$. This could be due to the fact that $\mathcal{H}$ is constructed by having robust linguistic literature supporting our hypothesis.

5.2 Second Experiment Comparing Three Different Alignments

The idea behind this experiment is to compare two pre-trained diachronic embeddings (presented in the subsection 5.1.1) by aligning them following three different approaches, deeper explained in the section 4.2:

- The classic alignments A_1 associated to the matrix $\mathbf{W}_1$ found in literature ((Hamilton et al., 2016; Conneau et al., 2017)), obtained by solving the Procrustes problem with orthogonal constrains.
- The alignment A_2 associated to $\mathbf{W}_2$, computed by solving an optimization problem based on a linear decomposition.
- The alignment A_3 associated to $\mathbf{W}_3$, computed by solving an optimization problem with 0-norm by applying the Frank-Wolfe method.

5.2.1 Esperimental setup

Alignment parameters In order to find the matrix $\mathbf{W}_3$ for the alignment A_3, we make some test by making varying the parameters of the Frank-Wolfe method. ϵ is made vary among $10^{-4}, 10^{-6}, 10^{-12}$, and g_k is initialized as $(g_0)_i$ by using a unit vector and then a randomly generated vector. The results seem to be quite similar for our purposes. Finally, we decide to fix $\epsilon = 10^{-12}$, and to take a randomly generated $(g_0)_i$ for the first iteration of Frank-Wolfe method. We set $\alpha = 5$ for all experiments.

5.2.2 Evaluation methodology

After applying the three alignments thorugh $\mathbf{W}_1$, $\mathbf{W}_2$, and $\mathbf{W}_3$, multiplying them by $\mathbf{X}$, the analysis of the results is structured into three parts. In the current subsection, let us indicate $e_{1890}(w) := \vec{x} \in \mathcal{V}_{1890}$ and $e_{1990}(w) := \vec{y} \in \mathcal{V}_{1990}$.

First part: a general evaluation on the test sets Given all the words $w \in \mathcal{H}$, $\mathcal{B}$, $\mathcal{S}$ we compute the mean value of $\mathfrak{s}_i(w)$ and the mean value of $\theta(\mathbf{W}_i\vec{x}, \vec{y})$ for $i = 1, 2, 3$.

Second part: evaluation of the top and bottom of the rankings $\mathfrak{r}_i$ We retrieve the rankings $\mathfrak{r}_1$, $\mathfrak{r}_2$, and $\mathfrak{r}_3$, by sorting $w \in \mathcal{D}$ with respect to the values $\mathfrak{s}_1$, $\mathfrak{s}_2$, and $\mathfrak{s}_3$ respectively in ascending order. Firstly, we take the first and the last 50 words of sorted lists $\mathfrak{r}_i$, obtaining the sets $\mathcal{L}_i$ and $\mathcal{H}_i$, $i = 1, 2, 3$:

- $\mathcal{L}_i$ are the sets made up by the first 50 words in $\mathfrak{r}_i$, associated with lower $\mathfrak{s}_i$ values. The words $w \in \mathcal{L}_i$ correspond to representations $e_{1890}(w)$ and $e_{1990}(w)$ which change more before and after the alignment through $\mathbf{W}_i$.

- $\mathcal{H}_i$ are the sets made up by the last 50 words in $\mathfrak{r}_i$, associated with higher $\mathfrak{s}_i$ values. The words $w \in \mathcal{H}_i$ correspond to representations $e_{1890}(w)$ and $e_{1990}(w)$ which change less before and after the alignment through $\mathbf{W}_i$.

This distinction is summarized in the table 5.7, where we report our corresponding interpretation concerning **cultural semantic conditioning**.

Set	$\mathfrak{s}_i$	Our hypothesis
$\mathcal{L}_i$	Lowest 50 $\mathfrak{s}_i$ values	More likely to be subject of **cultural semantic conditioning** according to the alignemnt through $\mathbf{W}_i$
$\mathcal{H}_i$	Highest 50 $\mathfrak{s}_i$ values	Less likely to be subject of **cultural semantic conditioning** according to the alignemnt through $\mathbf{W}_i$

Table 5.7: A table summarizing the properties of the elements in $\mathcal{L}_i$ and $\mathcal{H}_i$, and the corresponding hypothesis.

Then, we find the intersection among the test sets $\mathcal{H}$, $\mathcal{B}$, $\mathcal{S}$ and $\mathcal{L}_i$ and $\mathcal{H}_i$. We provide the mean values of the similarity measures $\mathfrak{s}_i(w)$ with $i = 1, 2, 3$, and $\mathbf{s}_E(w) := \mathbf{s}_E(\vec{w}_{1890}, \vec{w}_{1990})$ obtained by using the tools of the website Exploring Word Evolution Jatowt et al. ((2018)) described in the section 3.2.2. Moreover, we compute the averages of $\theta(\mathbf{W}_i\vec{x}, \vec{y})$, and $\theta(\vec{x}, \vec{y})$, with $w \in \mathcal{L}_i$ and then with $w \in \mathcal{H}_i$ for $i = 1, 2, 3$. Finally, we investigate the presence of semantic or syntactic regularities in $\mathcal{L}_i$ and $\mathcal{H}_i$.

Third part: comparison of the rankings through $\mathcal{H}$ In this last part of analysis, we focus on the single test set $\mathcal{H}$, by identifying the position of its elements in the rankings $\mathfrak{r}_1$, $\mathfrak{r}_2$, and $\mathfrak{r}_3$. More specifically, letting $i_i(w)$ be the index of the word w in $\mathfrak{r}_i$ with $i = 1, 2, 3$, we identify the words $w \in \mathcal{H}$ for which the absolute difference of indices $|i_i(w) - i_j(w)|$ $(i, j = 1, 2, 3, i < j)$ is higher than the threshold m. In that cases, fixing $p > 0$ we individuate three possibilities:

- $i_i(w) < i_j(w)$, $i_i(w) \leq p \Rightarrow$ the ranking $\mathfrak{r}_i$ classify w better than $\mathfrak{r}_j$.
- $i_j(w) < i_i(w)$, $i_j(w) \leq p \Rightarrow$ the ranking $\mathfrak{r}_j$ classify w better than $\mathfrak{r}_i$.
- Otherwise, $i_i(w)$ and $i_j(w)$ are close, or they are both far from the tops of the rankings $\mathfrak{r}_i$ and $\mathfrak{r}_j$. This could mean that their placement is realized according to our prediction or not in both rankings.

It is important to observe that the goal of this test is **to compare the alignments**. Finally, we make a qualitative analysis of the retrieved words. We decide to fix $m = 2000$ and $p = 1000$.

5.2.3 Results

First analysis: a general evaluation on the test sets For all the words $w \in \mathcal{D}$ in the sets $\mathcal{H}$, $\mathcal{B}$, and $\mathcal{S}$ we compute $\mathfrak{s}_1(w)$, $\mathfrak{s}_2(w)$, and $\mathfrak{s}_3(w)$ defined as in the section 4.3, where $\vec{y} \in \mathcal{V}_{1990}$ and $\vec{x} \in \mathcal{V}_{1890}$ are the representations of w in the two embeddings. In the following table 5.8 the averages of the measures $\mathfrak{s}_i(w)$ with the corresponding variances $\sigma_{\mathfrak{s}_i}$ with $i = 1, 2, 3$ in the three different sets are reported.

Set	Mean $\mathfrak{s}_1 \pm \sigma_{\mathfrak{s}_1}$	Mean $\mathfrak{s}_2 \pm \sigma_{\mathfrak{s}_2}$	Mean $\mathfrak{s}_3 \pm \sigma_{\mathfrak{s}_3}$
$\mathcal{H}$	0.04120 ± 0.00109	0.56898 ± 0.02608	$0{,}5581 \pm 0.02513$
$\mathcal{B}$	0.05414 ± 0.00185	0.62243 ± 0.01828	0.60922 ± 0.01905
$\mathcal{S}$	0.04562 ± 0.00109	0.66787 ± 0.00903	0.6518 ± 0.00918

Table 5.8: Table showing the averages of the measures $\mathfrak{s}_i$ with the corresponding variances $\sigma_{\mathfrak{s}_i}$ with $i = 1, 2, 3$ in $\mathcal{H}$, $\mathcal{B}$, and $\mathcal{S}$.

In table 5.8 it is possible to notice that while the first alignment A_1 leads to very low and similar $\mathfrak{s}_1$ values, both A_2 and A_3 recognize a difference of similarities $\mathfrak{s}_2$ and $\mathfrak{s}_3$ among the test sets. According to both alignments, the test set $\mathcal{H}$ of the words subject to historic semantic change corresponds to $\mathfrak{s}_i$ $(i = 2, 3)$ 0.1 higher than $\mathcal{S}$ of presumably stable words. Moreover, the mean $\mathfrak{s}_3$ of the biased words in $\mathcal{B}$ is lower than the mean $\mathfrak{s}_2$, which means that A_3 seems to be more sensitive to the semantic differences among the words of such test set.
Then, in table 5.9 the averages of $\theta(\mathbf{W}_1\vec{x}, \vec{y})$, $\theta(\mathbf{W}_2\vec{x}, \vec{y})$, and $\theta(\mathbf{W}_3\vec{x}, \vec{y})$ are reported, with the corresponding variances $\sigma_{\theta(\mathbf{W}_i\vec{x}, \vec{y})}$, with $i = 1, 2, 3$.

Set	Mean $\theta(\mathbf{W}_1\vec{x}, \vec{y}) \pm \sigma_{\theta(\mathbf{W}_1\vec{x}, \vec{y})}$	Mean $\theta(\mathbf{W}_2\vec{x}, \vec{y}) \pm \sigma_{\theta(\mathbf{W}_2\vec{x}, \vec{y})}$	Mean $\theta(\mathbf{W}_3\vec{x}, \vec{y}) \pm \sigma_{\theta(\mathbf{W}_3\vec{x}, \vec{y})}$
$\mathcal{H}$	0.02075 ± 0.01853	0.38679 ± 0.07001	0.38962 ± 0.07144
$\mathcal{B}$	0.12812 ± 0.10874	0.55 ± 0.07218	0.5625 ± 0.07078
$\mathcal{S}$	$0.00096 \pm 4.71523 \times 10^{-5}$	0.58557 ± 0.02128	0.57307 ± 0.02071

Table 5.9: Table showing the averages of the measures $\theta(\mathbf{W}_i\vec{x}, \vec{y})$ with the corresponding variances $\sigma_{\theta(\mathbf{W}_i\vec{x}, \vec{y})}$ with $i = 1, 2, 3$ in $\mathcal{H}$, $\mathcal{B}$, and $\mathcal{S}$.

From table 5.9 it is possible to evince that the average $\theta(\mathbf{W}_1\vec{x}, \vec{y})$ of each test set show a much higher change among words of $\mathcal{S}$ than the other sets (corresponding to a very low mean $\theta(\mathbf{W}_1\vec{x}, \vec{y}) = 0.00096$), which contradicts the hypothesis on the test sets, confirmed by the first experiment (section 5.1). This means that the alignment A_1 does not perceive the semantic change which corresponds to the validated test sets. On the other hand, the averages of $\theta(\mathbf{W}_2\vec{x}, \vec{y})$ and $\theta(\mathbf{W}_3\vec{x}, \vec{y})$ among test sets are similar and respect the hypothesis we do on such sets: $\mathcal{H}$ and $\mathcal{B}$ correspond to lower values than $\mathcal{S}$, even if the the difference il slight between $\mathcal{B}$ and $\mathcal{S}$.

Results of the second analysis: evaluation of the top and bottom of the rankings As explained formerly in the subsection 5.2.2, in this part of the analysis the rankings $\mathfrak{r}_1$, $\mathfrak{r}_2$, and $\mathfrak{r}_3$ are built, by sorting all the words in $\mathcal{D}$ in ascending order based on the values $\mathfrak{s}_1$, $\mathfrak{s}_2$, and $\mathfrak{s}_3$ respectively. Then, we construct $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_3$ taking the first 50 words of the rankings, and $\mathcal{H}_1$, $\mathcal{H}_2$, and $\mathcal{H}_3$ taking the last 50 words of the rankings. The sets $\mathcal{L}_i$ are shown in the table 5.11, while $\mathcal{H}_i$, are shown in 5.12, for $i = 1, 2, 3$.
The intersection among all the test sets $\mathcal{H}$, $\mathcal{B}$, $\mathcal{S}$ are shown in the table 5.10. All the words in $\mathcal{L}_i$, $\mathcal{H}_i$, with $i = 1, 2, 3$ also belonging to the test sets are in bold in the tables 5.11 and 5.12.
In the table 5.10 it is possible to observe that the intersections among test sets and the subsets of the ranking $\mathfrak{r}_1$ are not very significant, since there is just one word of $\mathcal{H}$ in $\mathcal{L}_1$ and one word of $\mathcal{B}$ in $\mathcal{H}_1$. On the contrary, we note a more positive trend corresponding to the subsets of the other two rankings $\mathfrak{r}_2$ and $\mathfrak{r}_3$. Indeed, both in $\mathcal{L}_2$ and $\mathcal{L}_3$ there are 3 words of $\mathcal{H}$ and $\mathcal{B}$, the test sets corresponding to the words more likely to be subject to semantic cultural conditioning. In $\mathcal{H}_2$ and $\mathcal{H}_3$ there are respectively 4 and 5 presumably stable words of $\mathcal{S}$, respecting our precision. However, in both the latter sets there are few words of $\mathcal{H}$, and none of $\mathcal{B}$.

$\cap$	$\mathcal{L}_1$	$\mathcal{H}_1$	$\mathcal{L}_2$	$\mathcal{H}_2$	$\mathcal{L}_3$	$\mathcal{H}_3$
$\mathcal{H}$	1	0	2	2	2	1
$\mathcal{B}$	0	1	1	0	1	0
$\mathcal{S}$	0	0	0	5	0	4

Table 5.10: Intersection among $\mathcal{H}$, $\mathcal{B}$, $\mathcal{S}$ and $\mathcal{L}_i$, $\mathcal{H}_i$, with $i = 1, 2, 3$

$\mathcal{L}_1$	$\mathcal{L}_2$	$\mathcal{L}_3$
kill, *cavalry, to, root, ought, introducing, representatives, distress, slumber, court, brow, greet, helpless, warmly, searched, unnoticed, talks, prolonged, concerned, conversing, devote, sideways, dobbin, convince, nodding, loom, brussels, clarence, centuries, firing, revolver, simple, signs, dimly, seized, mancha, kitchen, forest, rank, breath, somehow, grandmother, finishing, colors, aimed, arguments, console, stores, glance, everything*	***guy***, *dow, tony, quentin, gavin, georgie, vis, egyptian,* ***checking***, *mac, ut, barnaby, jan, the, dinah,* ***gypsy***, *cranford, comer, catriona, percy, deborah, dad, gilbert, jessie, palmer, jason, kenneth, martha, barker, edith, foster, nanny, ta, overdue, scot, almayer, comers, 66, covers, esther, signor, judy, headed, romances, clennam, tho, hilda, urge, amy, jean*	***guy***, *tony, quentin, gavin, egyptian, georgie, dow, mac, vis,* ***checking***, *ut, cranford, jan, the, dinah, barnaby,* ***gypsy***, *deborah, comer, jason, nanny, catriona, gilbert, jessie, dad, percy, edith, palmer, martha, esther, romances, foster, signor, almayer, barker, 66, overdue, ta, comers, dale, kenneth, amy, minstrel, covers, clennam, hilda, jenny, headed, weaving, joan*

Table 5.11: Elements of $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_3$, with first 50 words in $\mathfrak{r}_i$, associated with lower $\mathfrak{s}_i$ values, sorted in ascending order. The elements of the test sets $\mathcal{H}$, $\mathcal{B}$, and $\mathcal{S}$ are in bold.

$\mathcal{H}_1$	$\mathcal{H}_2$	$\mathcal{H}_3$
disadvantage, churches, impulse, shoulders, poured, sugar, immortal, assuring, retaining, gale, choosing, pew, happily, corresponding, income, relationship, bee, shudder, exquisite, jump, swam, exercised, overboard, waked, purchased, expressions, modest, quitted, tender, indignantly, recalling, couple, architecture, another, folded, plight, den, ***girl****, developed, went, burgundy, scornfully, blank, alternative, appealed, operations, influence, strengthened, vote, extinguished*	*marhaus, gaheris, beaumains, heard, sat, hight, trees, gat,* ***door****, miles, evening, balin,* ***water****, five, launcelot, jesu,* ***windows****, years, hear, ye, floor, church, hundred, voice, stairs, sea, kenwigs, steps, glass, unto, oh,* ***knight****, twenty, warm,* ***chair****, mother, hair, morning, standing, brownlow, gude, ried, thou, sitting,* ***window****, eyes, horses, hours, afternoon,* ***horse***	*marhaus, beaumains, gaheris, hight, heard, trees, gat, sat, miles, kenwigs, floor, balin,* ***door****, launcelot,* ***water****, years, hundred, evening, jesu, stairs, unto, ye,* ***windows****, church, sea, oh, voice, mother, five,* ***chair****, gude, twenty, sitting, thou, steps, hear, hair, glass, eight, sister, sit, standing, smile, dressed, feet, brownlow,* ***horse****, flowers, stared, hours*

Table 5.12: Elements of $\mathcal{H}_1$, $\mathcal{H}_2$, and $\mathcal{H}_3$, with the last 50 words in $\mathfrak{r}_i$, associated with higher $\mathfrak{s}_i$ values, sorted in descending order. The elements of the test sets $\mathcal{H}$, $\mathcal{B}$, and $\mathcal{S}$ are in bold.

For each of the 6 sets of words, we provide the mean $\mathfrak{s}_i$ value and the corresponding variance $\sigma_{\mathfrak{s}_i}$, based on our data, and the mean $\mathbf{s}_E(w)$ value and the corresponding variance $\sigma_{\mathbf{s}_E}$. The results are reported in the table 5.13 and in charts of the figure 5.2.

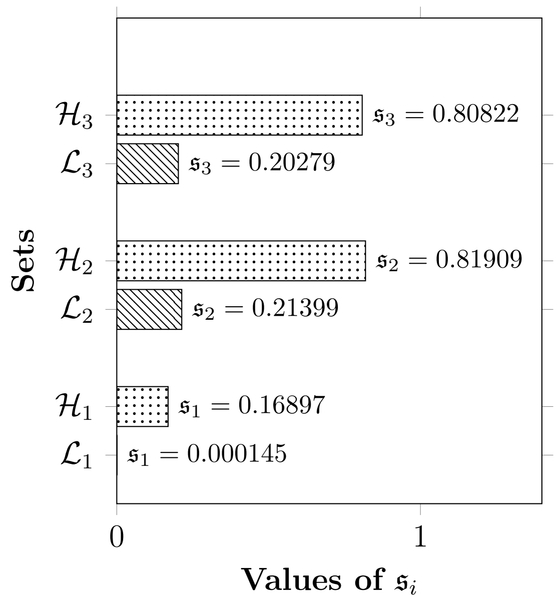
(a) $\mathfrak{s}_i$ values, with $i = 1, 2, 3$

Set	Mean $\mathfrak{s}_1 \pm \sigma_{\mathfrak{s}_1}$
$\mathcal{L}_1$	$0.000145 \pm 5.9 \times 10^{-9}$
$\mathcal{H}_1$	0.16897 ± 0.000139
	Mean $\mathfrak{s}_2 \pm \sigma_{\mathfrak{s}_2}$
$\mathcal{L}_2$	0.21399 ± 0.00517
$\mathcal{H}_2$	0.81909 ± 0.0004
	Mean $\mathfrak{s}_3 \pm \sigma_{\mathfrak{s}_3}$
$\mathcal{L}_3$	0.20279 ± 0.00524
$\mathcal{H}_3$	0.80822 ± 0.00039

(b) $\mathbf{s}_E$ values

Set	Mean $\mathbf{s}_E \pm \sigma_{\mathbf{s}_E}$
$\mathcal{L}_1$	0.9109 ± 0.00836
$\mathcal{H}_1$	0.88498 ± 0.02846
$\mathcal{L}_2$	0.55203 ± 0.10113
$\mathcal{H}_2$	0.95848 ± 0.00370
$\mathcal{L}_3$	0.566 ± 0.10532
$\mathcal{H}_3$	0.95529 ± 0.0037

Table 5.13: The averages of the corresponding variances of the similarity values in $\mathcal{L}_i$ and $\mathcal{H}_i$, $i = 1, 2, 3$



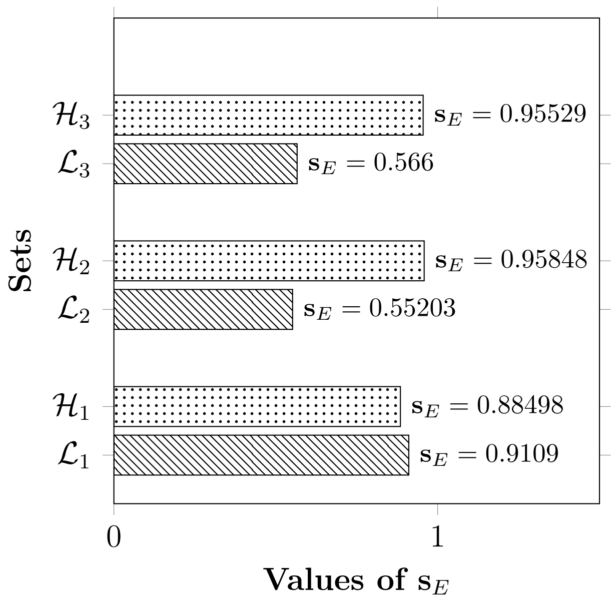


Figure 5.2: Bar charts representing the averages of the similarities $\mathfrak{s}_i$ and $\mathbf{s}_E$ of the sets $\mathcal{L}_i$ and $\mathcal{H}_i$, $i = 1, 2, 3$.

In the table 5.13(a), it is clear that the mean $\mathfrak{s}_1$ is overall low in both $\mathcal{L}_1$ and $\mathcal{H}_1$, which gathers the highest values with an average of only 0.16897. On the other hand, the mean $\mathfrak{s}_2$ and $\mathfrak{s}_3$ vary more, obtaining a 0.6 difference between the pairs $\mathcal{L}_i$ and $\mathcal{H}_i$ ($i = 2, 3$), corresponding to a more marked semantic difference in both cases.
In the table 5.13(b), it is possible to notice that while $\mathcal{L}_1$ and $\mathcal{H}_1$ correspond

to almost equally high mean values of $\mathbf{s}_E$, $\mathcal{L}_2$ and $\mathcal{L}_3$ seem to correspond to lower values of $\mathbf{s}_E$, and $\mathcal{H}_2$ and $\mathcal{H}_3$ to higher values of $\mathbf{s}_E$.
In summary, the similarity perceived through the alignment A_1 is not reflected by the other measure $\mathbf{s}_E$, while the same measure apparently confirm the diachronic similarity found according to the other alignments A_2, A_3.

Then, we compute the averages of $\theta(\vec{x}, \vec{y})$ with $\vec{x} \in \mathcal{V}_{1890}$ and $\vec{y} \in \mathcal{V}_{1990}$ corresponding to the same word w in the sets $\mathcal{L}_i$, $\mathcal{H}_i$, with $i = 1, 2, 3$. In addition, we calculate the mean values and the corresponding variance of $\theta(\mathbf{W}_i\vec{x}, \vec{y})$ for $\mathcal{L}_i$ and $\mathcal{H}_i$, for $i = 1, 2, 3$. All the results are reported in the table 5.14 and in the graphs of the figure 5.3.

(a) $\theta(\vec{x}, \vec{y})$ values

Set	Mean $\theta(\vec{x}, \vec{y}) \pm \sigma_{\theta(\vec{x},\vec{y})}$
$\mathcal{L}_1$	0.40399 ± 0.02388
$\mathcal{H}_1$	0.349 ± 0.02634
$\mathcal{L}_2$	0.088 ± 0.00195
$\mathcal{H}_2$	0.625 ± 0.02272
$\mathcal{L}_3$	0.091 ± 0.00217
$\mathcal{H}_3$	0.627 ± 0.02252

(b) $\theta(\mathbf{W}_1\vec{x}, \vec{y})$ values, with $i = 1, 2, 3$

Set	Mean $\theta(\mathbf{W}_1\vec{x}, \vec{y}) \pm \sigma_{\theta(\mathbf{W}_1\vec{x},\vec{y})}$
$\mathcal{L}_1$	0.002 ± 9.6×10^{-5}
$\mathcal{H}_1$	0.027 ± 0.00172
	Mean $\theta(\mathbf{W}_2\vec{x}, \vec{y}) \pm \sigma_{\theta(\mathbf{W}_2\vec{x},\vec{y})}$
$\mathcal{L}_2$	0.05 ± 0.0058
$\mathcal{H}_2$	0.796 ± 0.00888
	Mean $\theta(\mathbf{W}_3\vec{x}, \vec{y}) \pm \sigma_{\theta(\mathbf{W}_3\vec{x},\vec{y})}$
$\mathcal{L}_3$	0.05 ± 0.0054
$\mathcal{H}_3$	0.807 ± 0.0066

Table 5.14: The averages and the corresponding variances of the θ values in $\mathcal{L}_i$ and $\mathcal{H}_i$, $i = 1, 2, 3$

In the table 5.14(a) it is possible to observe that the mean $\theta(\vec{x}, \vec{y})$ in $\mathcal{L}_1$ is even higher than in $\mathcal{H}_1$, which means that there is not even a correspondence between the measures $\mathfrak{s}_1$ and θ. On the contrary, $\mathfrak{s}_2$ and $\mathfrak{s}_3$ seems to be coherent to θ according to A_2 and A_3 respectively. Indeed, in both pairs of subsets $\mathcal{L}_i$ and $\mathcal{H}_i$ with $i = 2, 3$, there is the same considerable difference between the mean $\theta(\vec{x}, \vec{y})$ values: in particular,the averages are much higher in $\mathcal{H}_2$ and $\mathcal{H}_3$. In the table 5.14(b) a similar trend emerges: $\theta(\mathbf{W}_2\vec{x}, \vec{y})$ and $\theta(\mathbf{W}_3\vec{x}, \vec{y})$ are positively unbalanced in the same way as before. On the other hand, this time $\theta(\mathbf{W}_1\vec{x}, \vec{y})$ is more coherent with $\mathfrak{s}_1$, but the corresponding values are very slow.

The next sep is the semantic and syntactic **regularities detection** among $\mathcal{L}_i$ and $\mathcal{H}_i$, with $i = 1, 2, 3$. Apparently, in the two subsets $\mathcal{L}_1$ and $\mathcal{H}_1$ extracted from the ranking $\mathfrak{r}_1$ there would not appear to be any regularity,

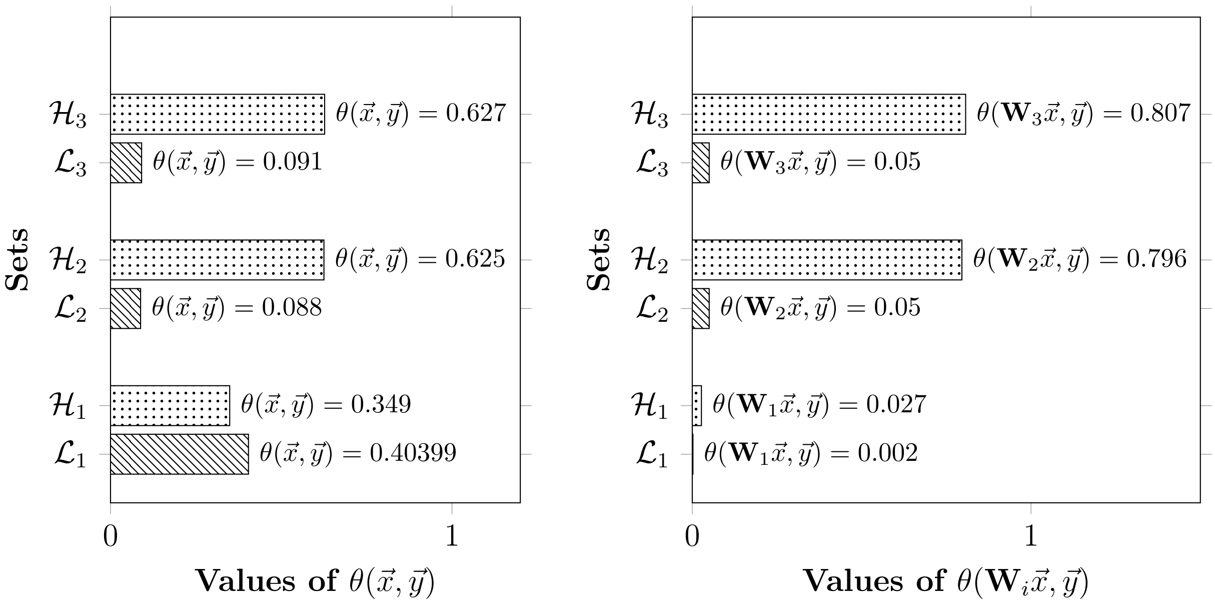


Figure 5.3: Bar chart representing the averages of $\theta(\vec{x},\vec{y})$ and $\theta(\mathbf{W}_i\vec{x},\vec{y})$ of the sets $\mathcal{L}_i$ and $\mathcal{H}_i$ with $i = 1, 2, 3$.

given that such words belongs to different semantic and syntactic categories. The respective mean $\mathfrak{s}_1$ values are low, while the mean $\mathfrak{s}_E$ values are overall high, showing a non-significant general semantic change.

On the contrary, the subsets $\mathcal{L}_2$, $\mathcal{H}_2$, and $\mathcal{L}_3$, $\mathcal{H}_3$ extracted from $\mathfrak{r}_2$ and $\mathfrak{r}_3$ respectively show a significant semantic and syntactic **imbalance**. Exactly 23 of the words of $\mathcal{L}_2$ (*tony, quentin, gavin, georgie, barnaby, jan, dinah, cranford, catriona, percy, deborah, gilbert, jessie, jason, kenneth, martha, edith, esther, judy, clennam, hilda, amy, jean*), and 23 of the words in $\mathcal{L}_3$ (coinciding with the ones of $\mathcal{L}_2$, except for *judy, jean* which are replaced by *jenny, joan*) are **proper names**, largely fallen out of use during the 1900s. In addition, 7 of the words of $\mathcal{L}_2$ (*guy, gypsy, comer, barker, foster, nanny, signor*) and 8 words of $\mathcal{L}_3$ (coinciding with the ones in $\mathcal{L}_2$, with the addition of *minstrel*) are *non proper names referred to people*, while none of the elements of $\mathcal{L}_2$ and $\mathcal{L}_3$ belongs to the categories of *inanimate objects* or *body parts and human attributes.* Just one words in $\mathcal{L}_3$ (*dale*) is in the category *natural elements and common animals.* It is interesting to observe the word *mac* is in both $\mathcal{L}_2$ and $\mathcal{L}_3$: probably its old meaning of *a familiar term of address to a man or boy whose name is not known to the speaker* originated in 1600s overlaps with the contemporary meanings of *Macintosh personal computer* or *short for macaroni*Dictionary ((2022b)). Moreover, we can notice that the word corresponding to the lowest $\mathfrak{s}_2$ and $\mathfrak{s}_3$ values is *guy*, which

belongs to the test set $\mathcal{H}$.
On the other hand, only 3 of the words in $\mathcal{H}_2$ and in $\mathcal{H}_3$ are proper names (*jesu, launcelot, brownlow*), but it is important to precise that they correspond to important religious, epic or literature characters. Moreover, 9 words in $\mathcal{H}_2$ (*gat, door, windows, window, floor, church, stairs, glass, chair*) and 8 words in $\mathcal{H}_3$ (coinciding with the ones of $\mathcal{H}_2$, excluding *window*) belong to the category *inanimate objects.* 4 words in both $\mathcal{H}_2$ and $\mathcal{H}_3$ (*trees, sea, water, horses* in $\mathcal{H}_2$, and *trees, sea, water, flowers* in $\mathcal{H}_3$) belong to the category *natural elements and common animals.* 3 elements of $\mathcal{H}_2$ (*hair, voice, eyes*) and 5 elements of $\mathcal{H}_3$ (*hair, voice, smile, feet, dressed*) belong to the category *body parts and human attributes.* 2 words of $\mathcal{H}_2$ (*knight, mother*) and 2 words of $\mathcal{H}_3$ (*mother*, *sister*) are *non proper names referred to people.* In the figure 5.4 a graph illustrates the distribution of the words of $\mathcal{L}_2$ and $\mathcal{H}_2$ in the mentioned categories. All these considerations are summarized in the table 5.15 and in the bar chart 5.4.

Category	$\mathcal{L}_2$	$\mathcal{H}_2$	$\mathcal{L}_3$	$\mathcal{H}_3$
Proper names	23	3	22	3
Non proper names referred to people	7	2	8	2
Body part and human attributes	0	3	0	5
Natural elements and common animals	0	4	1	4
Inanimate objects	0	9	0	9

Table 5.15: Table reporting the distribution of the words of $\mathcal{L}_2$, $\mathcal{H}_2$ and $\mathcal{L}_3$, $\mathcal{H}_3$ into the mentioned categories for the regularities detection.

Although the analysis is qualitative and subjective, since the choice of categories is arbitrary, it is clear that the distribution of some semantic and syntactic categories is related to the values $\mathfrak{s}_2$ and $\mathfrak{s}_3$, and consequently, depends on the position of the words in the rankings $\mathfrak{r}_2$ and $\mathfrak{r}_3$.

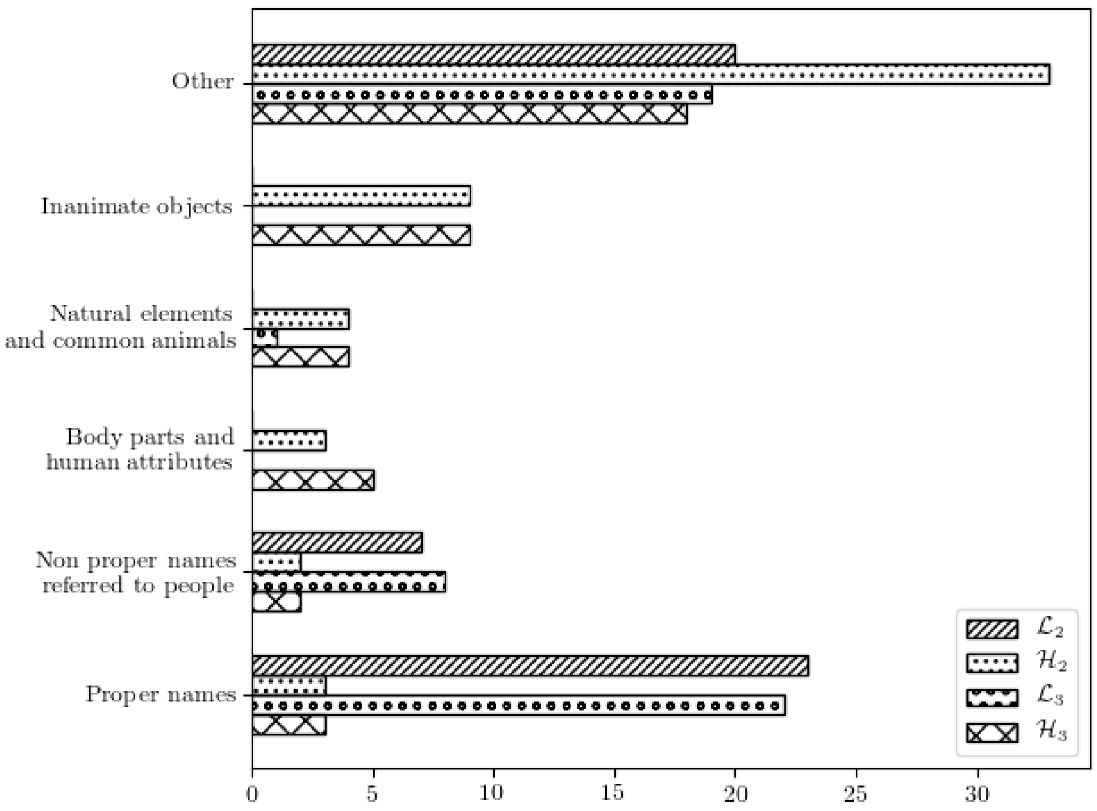


Figure 5.4: A graph representing the described regularities in $\mathcal{L}_2$ and $\mathcal{H}_2$, $\mathcal{L}_3$ and $\mathcal{H}_3$

Finally, we proceed to make a **qualitative comparison** of A_2 and A_3. Since $\mathcal{L}_2$ and $\mathcal{L}_3$ are quite similar, let us investigate more precisely the differences between them. In the tables 5.16 and 5.17 the intersections and the differences between such sets are shown. Focusing on the table of the differences 5.17, it is possible to notice that the alignment A_3 replaces in $\mathcal{L}_3$ the words *scot, judy, tho, urge, jean* of $\mathcal{L}_2$ with *dale, minstrel, jenny, weaving, joan*. While *jenny, joan* are proper nouns as well as *judy, jean*, we observe that *dale* and *minstrel* are expressions which are fallen into disuse throughout 20th century: indeed, according to the *Collins Dictionary*, *dale* is a term referring to *an open valley* in Old English ((Collins Dictionary, 2022a)), and *minstrel* is *a singer and musician who travelled around and entertained noble families in medieval times* ((Collins Dictionary, 2022b)).
Then, regarding $\mathcal{H}_2$ and $\mathcal{H}_3$, it is possible to observe a slight unbalanace. While in $\mathcal{H}_3$ there are 3 words of the category *body parts and human attributes* (*smile, feet, dressed*) in $\mathcal{H}_2$ there is only 1 (*eyes*). Moreover, contary

to $\mathcal{H}_3$, in $\mathcal{H}_2$ there is *window* belonging to the test set $\mathcal{H}$.

Intersection	$\mathcal{L}_2 \cap \mathcal{L}_3$	$\mathcal{H}_2 \cap \mathcal{H}_3$
Cardinality	45	42
Elements	*guy, dow, tony, quentin, gavin, georgie, vis, egyptian, checking, mac, ut, barnaby, jan, the, dinah, gypsy, cranford, comer, catriona, percy, deborah, dad, gilbert, jessie, palmer, jason, kenneth, martha, barker, edith, foster, nanny, ta, overdue, almayer, comers, 66, covers, esther, signor, headed, romances, clennam, hilda, amy*	*marhaus, gaheris, beaumains, heard, sat, hight, trees, gat, door, miles, evening, balin, water, five, launcelot, jesu, windows, years, hear, ye, floor, church, hundred, voice, stairs, sea, kenwigs, steps, glass, unto, oh, twenty, chair, mother, hair, standing, brownlow, gude, thou, sitting, hours, horse*

Table 5.16: Intersections between the pairs of sets $\mathcal{L}_i$ and $\mathcal{H}_i$, with $i = 2, 3$

Difference	$\mathcal{L}_3 - \mathcal{L}_2$	$\mathcal{H}_3 - \mathcal{H}_2$	$\mathcal{L}_2 - \mathcal{L}_3$	$\mathcal{H}_2 - \mathcal{H}_3$
Cardinality	5	8	5	8
Elements	*dale, minstrel, jenny, weaving, joan*	*eight, sister, sit, smile, dressed, feet, flowers, stared*	*scot, judy, tho, urge, jean*	*knight, warm, morning, tried, window, eyes, horses, afternoon*

Table 5.17: Differences between the pairs of sets $\mathcal{L}_i$ and $\mathcal{H}_i$, with $i = 2, 3$

Third part: comparison of the rankings through $\mathcal{H}$ We identify the words $w \in \mathcal{H}$ respecting the conditions $|i_i(w) - i_j(w)| > 2000, i_i \leq 1000$ or $i_j \leq 1000$, $i, j = 1, 2, 3$, $i < j$. Fixing $(i, j) = (2, 3)$, none word respects such conditions, and consequently we consider only $(i, j) = 1, 2$ and $(i, j) = 1, 3$. Moreover, also the comparison among the pairs $\mathfrak{r}_2$ and $\mathfrak{r}_3$ with respect to $\mathfrak{r}_1$ gives the same results, since, as shown before, $\mathfrak{r}_2$ and $\mathfrak{r}_3$ are similar. Hence, the considered sets of words, shown in the table 5.18 are:

- $w \in \mathcal{H}$ s.t. $i_1(w) - i_2(w) > 2000$, $i_1(w) - i_3(w) > 2000$, $i_2 \leq 1000$, $i_3 \leq 1000$, corresponding to the words better placed by $\mathfrak{r}_2$ and $\mathfrak{r}_3$ rather than $\mathfrak{r}_1$.

- $w \in \mathcal{H}$ s.t. $i_1(w) - i_2(w) < -2000$, $i_1(w) - i_3(w) < -2000$, $i_1 \leq 1000$, corresponding to the words better placed by $\mathfrak{r}_1$ rather than $\mathfrak{r}_2$ and $\mathfrak{r}_3$.

$w \in \mathcal{H}$ better placed by $\mathfrak{r}_2$ and $\mathfrak{r}_3$ than $\mathfrak{r}_1$	$w \in \mathcal{H}$ better placed by $\mathfrak{r}_1$ than $\mathfrak{r}_2$ and $\mathfrak{r}_3$
terrific, bully, gay, record, guy, checking, delivery, snap, toilet	*fun, tremendous, nice, kill, windows, instant, king*

Table 5.18: Table showing the selected words $w \in \mathcal{H}$ better placed by $\mathfrak{r}_2$ and $\mathfrak{r}_3$ rather than $\mathfrak{r}_1$ and viceversa.

In the table 5.18 it is possible to observe that the balance is slightly better for $\mathfrak{r}_2$ and $\mathfrak{r}_3$ in comparison to $\mathfrak{r}_1$, with 9 better placed words against 7 of $\mathfrak{r}_1$. Among them, it is important to notice that *instant* is a word related to social media language, much more recent than 1990, which means that its semantic change took placed after the epoch corresponding to the second embedding.

5.3 Discussion of the Results

The first experiment (section 5.1) prove that the social cultural conditioning, and expecially its declination of historic semantic change, takes actually place among diachronic embeddings trained on different corpora.
Regarding the second experiment (section 5.2), it emerges that overall, the alignments of our creation A_2 and A_3 give better results than the classic alignments A_1, which performs significantly worse in all our tests. This could be also due to the fact that in order to compute the alignment matrix $\mathbf{W}_1$ we consider all the words of the common dictionary to construct matrices $\mathbf{X}$ and $\mathbf{Y}$, and not the n most frequent ones, as shown in the section 1.4. Our choice is motivated by two reasons: first, the selection of n most frequent words used in two different epochs requires a deeper linguistic knowledge, which is beyond the scope of this thesis; second, we decide at the beginning to adopt the method used by Hamilton et al. ((2016)) as a reference.
Although A_2 and A_3 are in general more effective in highlighting cultural semantic conditioning, in the second part of the analysis conducted in the present chapter, A_3 apparently performs slightly better, expecially in the regularities detection. However, the present analysis can be expanded in many way, as we explain in **Conclusion and Future Works**.

Conclusion and Future Works

Word embeddings are built by training learning algorithms on large corpora of textual data, which often reflect different types of biases and cultural peculiarities inherited by the society itself. Since these tools are the state-of-the-art representations in NLP tasks, biases are likely to be carried over by Machine Learning algorithms, which may, in turn, reinforce them.
The aim of the present work is to build an alignment able to **highlight** the cultural semantic conditioning of word embeddings trained on different corpora. The alignment, a map between two different embedding space, is realized by multiplying the vectors $\vec{x}$ of the source space by the transformation matrix $\mathbf{W}$. Current alignments methods minimize the contribution to the error between $\mathbf{W}\vec{x}$ and the target vector $\vec{y}$ of every word representation. On the contrary, the hypothesis our work is based on is that the alignment should place the images of word vectors more distant to the target vector if they are more likely to be subject to cultural semantic conditioning, and viceversa. The first contribution is to model our hypothesis as a nonlinear optimization problem, whose resolution has been made possible by *ad hoc* **linear decompositions**. The more original contribution consists in the application of such decomposition to the Frank-Wolfe method, leading to a considerable reduction in computational cost. The experiments we conducted confirm our hypothesis: as an overall trend, the alignments we propose show an error proportional to the semantic change of a single words represented in two diachronic embeddings, trained on data 100 years apart.
However, the significant results of the present work can be seen as a springboard to **future works**. Indeed, it is possible to expand our research in the following directions:

- combining pairs of diachronic embeddings, beyond the decades 1890 and 1990, in order to investigate more generally how language evolves;
- training word embeddings on coeval data of different political ideology, or related to different cultures, places and social contexts, and then applying the same methodogical analysis on them;

- another option is making the regularities detection shown in the section 5.2.3 more systematic, by exploiting the ontology provided by Miller ((1998)) in the famous online project *Wordnet*;

- an other interesting idea could be to give sociological and linguistic interpretation to the trend of θ measure shown in figure 5.1.

Appendix A

In order to compute $\mathbf{W}_2$, the following AMPL code has been executed with the command:

```
include decomposition.run
```

with the file `decomposition.run` given by

```
reset;
model decomposition.mod;
data 1890_1990.dat;

option solver cplexamp;

for {it in 1..300} {
   let j:=it;
   solve;\newline
   }
```

which refers to the file `decomposition.mod`, whose script is

```
set I:=1..8810;
set K:=1..300;
set J:=1..300;

param j, in J, default 1;
param x{K,I};
param y{J,I};

var z{I};
var W{J,K};

```

```
minimize f: sum{i in I}z[i];
s.t. v{i in I}:
   sum{k in K} W[j,k]*x[k,i]-z[i] <= y[j,i];
s.t. v1{i in I}:
   sum{k in K} W[j,k]*x[k,i]+z[i] >= y[j,i];
```

Finally `1890_1990.dat` is a text document which contains the assignment of the parameters `x`, `y` (corresponding to the previously mentioned matrices **X** and **Y**), and `j`.

Appendix B

In order to compute $\mathbf{W}_3$, the following AMPL code has been executed with the command:

```
include FW.run
```

```
reset;

model FW.mod;
data 1890_1990.dat;.dat;
option solver cplexamp;

param z_curr{I};
param eps:=1e-12;
param it_fw;
param grad_mean:=5;
param grad_variance:=3.5;

let it_fw:=0;
let {i in I} grad_curr[i]:=
   max(Normal(grad_mean, grad_variance), 0);
let {i in I : grad_curr[i]>10}grad_curr[i]:=10;

for {it in 1..300}{
  let j:=it;
  solve;
}
let it_fw:=1;
repeat {
    let {i in I} z_curr[i]:=sum{k in K}a[k,i];
    let {i in I}grad_curr[i]:=5*exp(-5*z_curr[i]);
```

```
      for {it in 1..300} {
         let j:=it;
         solve;
  }
until sum{i in I}grad_curr[i]*(sum{k in K}a[k,i]+
   -z_curr[i])>=-eps;
```

referring to the file `FW.mod`, whose script is

```
set I:=1..8810;
set K:=1..300;
set J:=1..300;

param x{K,I};
param y{J,I};
param grad_curr{I};
param j, in J, default 1;

var a{K,I};
var W{J,K};

minimize f: sum{i in I}grad_curr[i]*a[j,i];
s.t. v1{i in I}:
  sum{k in K}W[j,k]*x[k,i]-a[j,i] <= y[j,i];
s.t. v2{i in I}:
  sum{k in K}W[j,k]*x[k,i]+a[j,i] >= y[j,i];
```

The reduced dimension version is given by the same `FW.run` file combined with a slightly different `.mod` file called `FW_RD.mod`, given by the following script:

```
set I:=1..8810;
set K:=1..300;
set J:=1..300;

param x{K,I};
param y{J,I};
param grad_curr{I};
param j, in J, default 1;

var a{K,I};
var W{J,K};

minimize f:
   sum{i in I : z_curr[i]<>0}grad_curr[i]*a[j,i];
s.t. v1{i in I : z_curr[i]<>0}:
  sum{k in K}W[j,k]*x[k,i]-a[j,i] <= y[j,i];
s.t. v2{i in I : z_curr[i]<>0}:
  sum{k in K}W[j,k]*x[k,i]+a[j,i] >= y[j,i];
s.t. v3{i in I: z_curr[i]=0}:
  sum{k in K}W[j,k]*x[k,i]-y[j,i]=0;
```

List of Figures

Bibliography

Zakaria Kaddari, Youssef Mellah, Jamal Berrich, and Mohammed G. Belkasmi. Studying Political Bias via Word Embeddings. In *WWW '20: Companion Proceedings of the Web Conference 2020*, volume 144, 2021.

Eva Picardi. *Le teorie del significato.* Editori Laterza, 1999.

Daniel Jurafsky and James Martin. *Speech and Language Processing.* Prentice Hall, 2000.

Zi Yin and Yuanyuan Shen. On the Dimensionality of Word Embedding. *Advances in neural information processing systems*, 31, 2018.

Charles E. Osgood, George J. Suci, and Percy Tannenbaum. *The Measurement of Meaning.* University of Illinois press, 1957.

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

J. Firth. A Synopsis of Linguistic Theory 1930-1955. In *Studies in Linguistic Analysis.* Philological Society, Oxford, 1957.

Martin Joos. Description of Language Design. *The Journal of the Acoustical Society of America*, 22(6):701–707, 1950.

Burghard B Rieger. *On Distributed Representation in Word Semantics.* International Computer Science Institute Berkeley, CA, 1991.

Gema Atienza. Operations on Word Vectors. `https://github.com/gemaatienza/Deep-Learning-Coursera`, 2018.

David E. Rumelhart and Adele A. Abrahamson. A Model for Analogical Reasoning. *Cognitive Psychology*, 5(1):1–28, 1973.

Tomas Mikolov, Wen tau Yih, and Geoffrey Zweig. Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 746–751, 2013a.

Omer Levy, Yoav Goldberg, and Ido Dagan. Improving Distributional Similarity with Lessons Learned from Word Embeddings. *Transactions of the association for computational linguistics*, 3:211–225, 2015a.

Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. Dive into Deep Learning. *arXiv preprint arXiv:2106.11342*, 2021.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word representations in Vector Space. *arXiv preprint arXiv:1301.3781*, 2013b.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Medium - Word Embedding and One Hot Encoding. https://medium.com/intelligentmachines/word-embedding-and-one-hot-encoding-ad17b4bbe111#, 2020.

Shashank Gupta. Word Vector Encoding in NLP (Make Machine Understand Text). https://www.enjoyalgorithms.com/blog/word-vector-encoding-in-nlp, 2022.

Saif Shabou. Natural Language Processing with R - Chapter 3, Section 2. https://s-ai-f.github.io/Natural-Language-Processing/Word-embeddings.html#one-hot-encoding, 2020.

Bhoomika Madhukar. The Continuous Bag Of Words (CBOW) Model in NLP – Hands-On Implementation With Codes. https://analyticsindiamag.com/the-continuous-bag-of-words-cbow-model-in-nlp-hands-on-implementation-with-codes/, 2020.

Stanford. CS229 - Syllabus and Course Schedule. http://cs229.stanford.edu/syllabus-spring2020.html, 2020.

Wikipedia. Cross Entropy. https://en.wikipedia.org/wiki/Cross_entropy, 2022a.

Sanket Doshi. Skip-Gram: NLP context words prediction algorithm. https://towardsdatascience.com/skip-gram-nlp-context-words-prediction-algorithm-5bbf34f84e0c, 2019.

Sketch Engine. POS tags. `https://www.sketchengine.eu/blog/pos-tags/#`, 2018.

Alexander Kalinowski and Yuan An. A Survey of Embedding Space Alignment Methods for Language and Knowledge Graphs. *arXiv preprint arXiv:2010.13688*, 2020.

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change. *arXiv preprint arXiv:1605.09096*, 2016.

Sebastian Ruder, Ivan Vulić, and Anders Søgaard. A Survey of Cross-Lingual Word Embedding Models. *Journal of Artificial Intelligence Research*, 65: 569–631, 2019.

Aditya Mogadala and Achim Rettinger. Bilingual Word Embeddings from Parallel and Non-parallel Corpora for Cross-Language Text Classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 692–702, San Diego, California, 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1083. URL `https://aclanthology.org/N16-1083`.

Peter Prettenhofer and Benno Stein. Cross-Language Text Classification Using Structural Correspondence Learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1118–1127, Uppsala, Sweden, July 2010. Association for Computational Linguistics. URL `https://aclanthology.org/P10-1114`.

Will Y. Zou, Richard Socher, Daniel Cer, and Christopher D. Manning. Bilingual Word Embeddings for Phrase-Based Machine Translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1393–1398, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL `https://aclanthology.org/D13-1141`.

Christian S. Perone, Roberto Silveira, and Thomas S. Paula. Evaluation of Sentence Embeddings in Downstream and Linguistic Probing Tasks. *arXiv preprint arXiv:1806.06259*, 2018.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. Exploiting Similarities among Languages for Machine Translation. *arXiv preprint arXiv:1309.4168*, 2013c.

Peter H Schönemann. A Generalized Solution of the Orthogonal Procrustes Problem. *Psychometrika*, 31(1):1–10, 1966.

Chao Xing, Dong Wang, Chao Liu, and Yiye Lin. Normalized word embedding and orthogonal transform for bilingual word translation. In *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1006–1011, 2015.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 2289–2294, 2016.

Samuel L. Smith, David H.P. Turban, Steven Hamblin, and Nils Y. Hammerla. Offline Bilingual Word Vectors, Orthogonal Transformations and the Inverted Softmax. *arXiv preprint arXiv:1702.03859*, 2017.

Manaal Faruqui and Chris Dyer. An information theoretic approach to bilingual word clustering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 777–783, 2013.

Angeliki Lazaridou, Georgiana Dinu, and Marco Baroni. Hubness and Pollution: Delving into Cross-Space Mapping for Zero-Shot Learning. In *Zong C, Strube M, editors. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers); 2015 Jul 26-31; Beijing, China. Stroudsburg (PA): Association for Computational Linguistics; 2015. p. 270-80.* ACL (Association for Computational Linguistics), 2015.

Milos Radovanovic, Alexandros Nanopoulos, and Mirjana Ivanovic. Hubs in Space: Popular Nearest Neighbors in High-Dimensional Data. *Journal of Machine Learning Research*, 11(sept):2487–2531, 2010.

Dimitris Bertsimas and John N. Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena scientific Belmont, MA, 1997.

Stephen Wright and Jorge Nocedal. Numerical Optimization. *Springer Science*, 35(67-68):7, 1999.

Least Absolute Deviation Regression, pages 299–302. Springer New York, New York, NY, 2008. ISBN 978-0-387-32833-1. doi: 10.1007/978-

0-387-32833-1_225. URL https://doi.org/10.1007/978-0-387-32833-1_225.

Marguerite Frank and Philip Wolfe. An Algorithm for Quadratic Programming. *Naval Research Logistics Quarterly*, 3, 1956.

Francesco Rinaldi, Immanuel M. Bomze, and Damiano Zeffiro. Frank–Wolfe and friends: a journey into projection-free first-order optimization methods. *4OR*, 2021.

Wikipedia2Vec. Pretrained Embeddings. https://wikipedia2vec.github.io/wikipedia2vec/pretrained/, 2020.

Davor Petreski and Ibrahim C. Hashim. Word embeddings are biased. But whose bias are they reflecting? *AI & SOCIETY*, pages 1–8, 2022.

OED Oxford English Dictionary. Bias. https://www.oed.com/view/Entry/18564?rskey=xGzzk8&result=1&isAdvanced=false#eid, 2021.

Kate Crawford. The trouble with bias. In *Proceedings of NeurIPS 2017*, 2017.

Batya Friedman and Helen Nissenbau. Bias in Computer Systems. *ACM Transactions on information systems (TOIS)*, 14(3):330–347, 1996.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. Language (Technology) is Power: A Critical Survey of" Bias" in NLP. *arXiv preprint arXiv:2005.14050*, 2020.

Orestis Papakyriakopoulos, Simon Hegelich, Juan Carlos Medina Serrano, and Fabienne Marco. Bias in Word Embeddings. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 446–457, 2020.

Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. Societal Biases in Language generation: Progress and Challenges. *arXiv preprint arXiv:2105.04054*, 2021.

Aylin Caliskan, Pimparkar Parth Ajay, Tessa Charlesworth, Robert Wolfe, and Mahzarin R. Banaji. Gender Bias in Word Embeddings: A Comprehensive Analysis of Frequency, Syntax, and Semantics. *arXiv preprint arXiv:2206.03390*, 2022.

Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam T. Kalai. Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. *Advances in Neural information processing systems*, 29, 2016.

Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. Norms of Valence, Arousal, and Dominance for 13,915 English lemmas. *Behavior Research Methods*, 45(4):1191–1207, 2013.

Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017.

Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*, 115(16):E3635–E3644, 2018.

Svetlana Kiritchenko and Saif M. Mohammad. Examining Gender and Race Bias in Two Hundred Sentiment Analysis Systems. *arXiv preprint arXiv:1805.04508*, 2018.

Thomas Davidson, Debasmita Bhattacharya, and Ingmar Weber. Racial bias in hate speech and abusive language detection datasets. *arXiv preprint arXiv:1905.12516*, 2019.

Josh Gordon, Marzieh Babaeianjelodar, and Jeanna Matthews. Studying Political Bias via Word Embeddings. In *WWW '20: Companion Proceedings of the Web Conference 2020*, pages 760–764, 2020.

Leonard Bloomfield. *Language.* George Allen & UNWIN LTD, 1933.

Collins Dictionary. diachronic. `https://www.collinsdictionary.com/dictionary/english/diachronic`, 2022a.

Yuri Lin, Jean-Baptiste Michel, Erez Aiden Lieberman, Jon Orwant, Will Brockman, and Slav Petrov. Syntactic Annotations for the Google Books NGram Corpus. In *Proceedings of the ACL 2012 system demonstrations*, pages 169–174, 2012.

GoogleBooks. The Google Books Ngram Viewer. `https://storage.googleapis.com/books/ngrams/books/datasetsv3.html`, 2013.

Mark Davies. Expanding horizons in historical linguistics with the 400-million word Corpus of Historical American English. *Corpora*, 7(2):121–157, 2012.

Omer Levy, Yoav Goldberg, and Ido Dagan. Improving Distributional Similarity with Lessons Learned from Word Embeddings. *Transactions of the association for computational linguistics*, 3:211–225, 2015b.

Adam Jatowt, Ricardo Campos, Sourav S. Bhowmick, Nina Tahmasebi, and Antoine Doucet. Every word has its history: Interactive exploration and visualization of word sense evolution. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1899–1902, 2018.

Mozhi Zhang, Keyulu Xu, Ken ichi Kawarabayashi, Stefanie Jegelka, and Jordan Boyd-Graber. Are Girls Neko or Shojo? Cross-Lingual Alignment of Non-Isomorphic Embeddings with Iterative Normalization. *arXiv preprint arXiv:1906.01622*, 2019.

Yerai Doval, Jose Camacho-Collados, Luis Espinosa-Anke, and Steven Schockaert. Improving Cross-Lingual Word Embeddings by Meeting in the Middle. *arXiv preprint arXiv:1808.08780*, 2018.

Anders Søgaard, Sebastian Ruder, and Ivan Vulić. On the Limitations of Unsupervised Bilingual Dictionary Induction. *arXiv preprint arXiv:1805.03620*, 2018.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. *arXiv preprint arXiv:1710.04087*, 2017.

Francesco Rinaldi, Marco Sciandone, and F. Shoen. Concave programming for minimizing the zero-norm over polyhedral sets. *Comput Optim Appl 46*, 2008.

Mental Floss. 13 Words That Changed From Negative to Positive Meanings (or Vice Versa). `https://www.mentalfloss.com/article/65987/13-words-changed-negative-positive-or-vice-versa`, 2015.

Wikipedia. Semantic change. `https://en.wikipedia.org/wiki/Semantic_change#`, 2022b.

Vivek Kulkarni, Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. Statistically Significant Detection of Linguistic Change. In *Proceedings of the 24th international conference on world wide web*, pages 625–635, 2015.

Adam Jatowt and Kevin Duh. A Framework for Analyzing Semantic Change of Words across Time. In *IEEE/ACM Joint Conference on Digital Libraries*, pages 229–238. IEEE, 2014.

Derry Tanti Wijaya and Reyyan Yeniterzi. Understanding Semantic Change of Words Over Centuries. In *Proceedings of the 2011 international workshop on DETecting and Exploiting Cultural diversiTy on the social web*, pages 35–40, 2011.

ONGIG. 25+ Examples of Biased Language. `https://blog.ongig.com/diversity-and-inclusion/biased-language-examples/`, 2020.

Collins Dictionary. mac. `https://www.collinsdictionary.com/dictionary/english/mac`, 2022b.

Collins Dictionary. dale. `https://www.collinsdictionary.com/dictionary/english/dale`, 2022a.

Collins Dictionary. minstrel. `https://www.collinsdictionary.com/dictionary/english/minstrel`, 2022b.

George A Miller. *WordNet: An electronic lexical database.* MIT press, 1998.