Words to Vectors and Back: A Survey on Cross-Lingual Word Embeddings

Aleena lifiya G00105151@AUS.EDU
MS Machine Learning
American University of Sharjah
University city , Sharjah , UAE

Abstract

Word embeddings are foundational to many natural language processing (NLP) tasks. Developing multilingual embedding spaces is crucial for seamless cross-lingual transfer and improved NLP capabilities. This survey provides an accessible introduction to cross-lingual word embeddings, focusing on the underlying mathematics.

The survey starts with key terminologies and the skip-gram model, a fundamental method for creating monolingual embeddings. It then examines the isomorphism hypothesis, an essential assumption in some alignment methods, before discussing various training methodologies, associated challenges, and evaluation metrics. Visualization technique t-SNE is also covered for analyzing embedding spaces. Finally open problems and future directions in this domain are discussed.

Keywords: Natural Language Processing, cross lingual word embedding, isomorphism hypothesis, training methods , mapping methods, unsupervised methods, joint training , evaluation metrics, T-SNE

1 Introduction

People communicate through technology and to technology more than ever. A lot of people are able to enjoy the benefits of technologies like a smart home assistant in their day to day life. But how does the grandma who speaks Malayalam use such a device?

Word embeddings are representation of words that capture their semantic meaning Rumelhart et al. (1986). (Mikolov et al., 2013b) introduced skip gram and CBOW (Continuous Bag of Words) that were able to create such embeddings efficiently giving rise to word2vec that was trained on a billion words and has the representation of 3 million words. (Mikolov et al., 2013a) then introduced mapping methods that can map one word embedding to other using a translational matrix. The key challenge to using mapping methods was the need for large amount of parallel data. In order to overcome it there has were several attempts : unsupervised methods (Lample et al., 2017)(Conneau et al., 2017) , semi-supervised methods (Wang et al., 2019) (Hangya et al., 2023-01-01) (Woller et al., November 1, 2021) (Eder et al., August 1, 2021). Several other methods were introduces in order to improve the structure of cross lingual word embeddings as discussed by (Ruder et al., 2019) .

Since this is an extremely broad topic , a newcomer who is interested in the domain might be overwhelmed with its breadth. Additionally , I believe understanding the mathematical machinery behind a concept can help us understand the topic better especially if the math is based on topics that we are already familiar with , conversely , one can understand math

better if it is tied to a more tangible topic that they can understand through different modalities like using code. Essentially I wanted a handbook like the (Boykis, 2023)that includes the essential tools without being overwhelming and hence this attempt.

The purpose of this survey is to provide a gentle introduction to cross-lingual word embedding for anyone interested in this domain. I aim to offer a cohesive guide through this expansive field, providing the reader with the foundational knowledge necessary to understand word embeddings. To achieve this, I have included explanations of the mathematical mechanisms wherever applicable. While many resources are available to understand these concepts, my goal here is to consolidate and summarize them in a coherent manner.

This survey is divided into nine sections, excluding the introduction. Sections 1 and 2 introduce basic terminology that the reader will encounter throughout the survey. In Section 3, the skip-gram method is discussed as a warm-up to word embeddings. Section 4 covers the isomorphism hypothesis, a key assumption for many training methods. Section 5 explores various training methods, while Section 6 addresses challenges and the factors influencing the quality of word embeddings. Section 7 reviews evaluation metrics, and Section 8 explains the working mechanism of t-SNE. Finally, Section 9 presents open problems and future directions in the field.

2 Some Terminologies

1. **Lexicon:** set of words in a language

2. **Seed lexicon:** set of words used to train a word embedding

3. **Cognates:** Two words are cognates if they are derived from a common language by the languages the words belong . Example : Theory in English and Théorie in French both are derived from ancient Greek word theoría

4. **Vocabulary:** Set of unique words used in a document or group of documents.

5. **Bilingual signal:** The data that is used by the model to learn to align a cross lingual representation space .

6. **Homographs**: Words that are spelled the same but different pronunciation and meanings and . For example 'lead' the metal and 'lead' the verb of 'leading'.

7. **Homophones:** Words that are pronounced the same but spelled differently and have different meanings. For example , in French : auteur means author and hauteur means height even though they are pronounced the same.

8. **Homonyms:** set of homophones and homographs

9. **parallel data**: set of word and its translation. Example: 'Apfel' in German and its English translation 'Apple'

10. **Low resource language:** Languages that are understudied and have scarce language resource digitized. Example : Kashmiri (Costa-jussà et al., 2022)

11. **Cross lingual transfer**: Use of a resource rich language to solve tasks in another , commonly lower resource languages, through transfer learning. An example of such task be multilingual voice agents.

3 Tokens,Vectors and Embeddings

In order to give text as input to a model , it needs to go through the following steps:

1. Tokenisation
2. Vectorisation
3. embedding

3.1 Tokens

Consider the following sentence:

`Advanced Machine Learning`

The question we are interested here is , how do we break this down into smaller units called *tokens*
Figure 1 shows the different type of tokens and

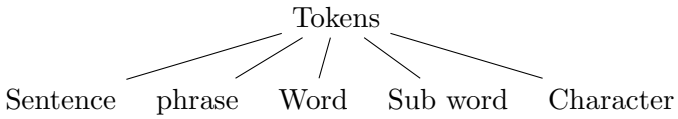


Figure 1: Different types of tokens

Despite some recent attempts on language independent tokenisation approaches (Bollegala et al., 2020) , it is considered a language specific problem. When a punctuation or white space delimiter would provide decent results for non agglutinative language like English , they cannot be used for languages that donot use white spaces between their words like Japanese.
For example ,`Advanced Machine Learning` is tokenised as `'advanced' , 'machine' , 'learning'` whereas in German, `Fortgeschrittenes Maschinelles Lernen` is tokenised as `'Fort', '##geschritten', '##es', 'Maschine', '##lle', '##s', 'Lernen'` reflective of its fusional nature.

Sentence
TEXT:

It has provided not only physical but also psychological sanctuary. It has been a guardian of identity. Over the years, its owners have returned from periods away and, on looking around them, remembered who they were.

TOKENS:
Token 1: 'It has provided not only physical but also psychological sanctuary.'

Token 2: 'It has been a guardian of identity.'
Token 3: 'Over the years, its owners have returned from periods away and, on looking around them, remembered who they were.'

Phrase
TEXT:

John Ruskin proposed that we seek two things of our buildings. We want them to shelter us. And we want them to speak to us – to speak to us of whatever we find important and need to be reminded of."

TOKENS:
Token 1: 'john ruskin proposed that we seek two things of our buildings.'
Token 2: 'we want them to shelter us.'
Token 3: 'we want them to speak to us'
Token 4: 'to speak to us of whatever we find important'
Token 5: 'need to be reminded of'

Word
TEXT:

architecture of happiness

TOKENS:
Token 1: 'architecture'
Token 2: 'of'
Token 3: 'happiness'

Sub word
TEXT:
"Lebe, wie du, wenn du stirbst, wünschen wirst, gelebt zu haben."
Tokens:
['Lebe', ',', 'wie', 'du', ',', 'wenn', 'du', 'st', '##ir', '##bst', ',', 'wünschen', 'wir', '##st', ',', 'gelebt', 'zu', 'haben', '.']

Character
TEXT:
'Haus' ['H','a','u','s]
Vectors are numerical representation of tokens. They can be thought of as point in an n dimensional space. These dimensions contain some semantic meaning. So word vector can help capture the semantic meaning of a word. The dimensionality of a word vector determines the nuances that can be captured by the vector. For example consider the word `dog`. Let it have a 3 dimensional vector representation `[0.9,-0.01,0.8]` for `['living','aquatic','mammal']`. A `cat` would have similar values in these dimensions. But if we increase the dimensions , we could capture more nuances that would differentiate a `dog` from a `cat` .

3.2 Embedding

How do we get from tokens to vectors that meaningfully capture semantic meanings? This is the result of the embedding process. The model (discussed in section 4 is trained on a large number of tokens and generates vectors that capture meaningful semantic relationship. The space shared by the vectors is the embedding space.

4 Distributed Representation of words

(Rumelhart et al., 1986) introduced the concept of distributed representation of words. In their word they build two isomorphic family trees , one with English names and other with Italian names. Then they built a model to predict the relationship a person has with another member in the same family. The first layer of their network had input that were the names of English people. The second layer was able to find a distributed representation of these words. So once the model was trained on the family tree of people with English names , they were able to use the same representation of names on their Italian equivalent due to the isomorphic nature of the two family trees. In this section we will discuss the skip-gram method which is a cornerstone approach in creating such scalable representations.

4.1 Skip-gram

The task associated with skip-gram (Mikolov et al., 2013b) is to find context words w_{t+j} around the given center word w_t . Formally, for given training words $\{w_1, w_2, w_3, \dots w_n\}$, skip-gram aims to maximize the average log probability:

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0} logp(w_{t+j}|w_t\} \tag{1}$$

This simply means that the model checks the probability that a word is a context word within a context window c around the given word w_t that is the center word. In order to understand the skip gram model lets look at a simple multi-layer perceptron with only one hidden layer in Figure 2.

The input layer is a one hot vector $\mathbf{w_I}$. $\mathbf{W_1}$ represents the weight matrix associated with the weights between the input layer and the hidden layer. The output layer describes the probabilities of the other words being the context word. $\mathbf{W_2}$ represents the weight matrix associated with the weights between the hidden layer and the output layer. Since each node of the input layer is connected to every node of the hidden layer, the dimension of $\mathbf{W_1}$will be $(300, 10000)$. After training, given a $\mathbf{w_I}$, when the dot product of this value and $\mathbf{W_1}$ is taken, we get a $(300, 1)$ vector $\mathbf{v_{w_I}}$ that is the word embedding of $\mathbf{w_I}$. The dot product of $\mathbf{v_{w_I}}$ and $\mathbf{W_2}$ is the input to the softmax layer. The dimensions of $\mathbf{W_2}$ is $(10000, 300)$. Thus, for each of the output node there will be a $(300, 1)$ vector $\mathbf{v_{w_O}}$.

The softmax then checks the probability of $\mathbf{v_{w_O}}$ being the representation of the context word for the word represented by $\mathbf{v_{w_I}}$. The skip-gram model thus defines the

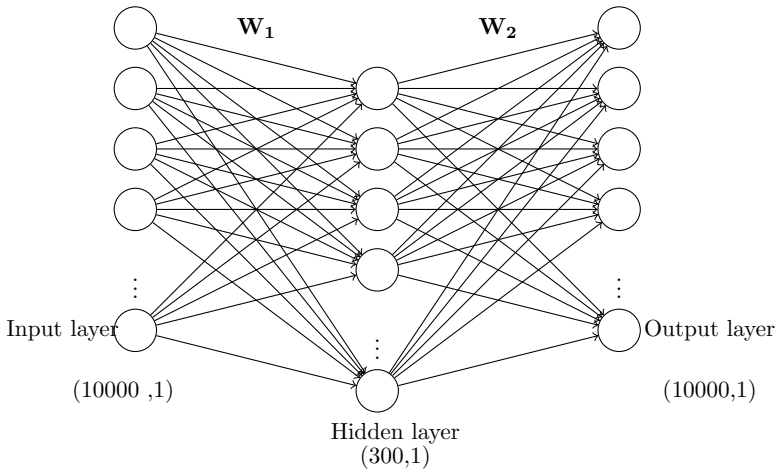


Figure 2: A multi-layer perceptron with one hidden layer. The input layer takes the one hot encoded values of word w_{t+1} , $\mathbf{W_1}$ is dotted with the input values to get the vector representation of the input values. $\mathbf{W_1}$ is dotted with the output hidden layer to get the vector representation of w_{t+1}

$log\ p\left(w_{t+j}|\ w_t\right)$ using the softmax function as:

$$p\left(w_O\ |w_I\right) = \frac{exp\big({\mathbf{v_{w_O}}}^\top \mathbf{v_{w_I}}\big)}{\sum_{w=1}^{W} exp({\mathbf{v_{w_O}}}^\top \mathbf{v_{w_I}})} \tag{2}$$

The downside to this approach is that it is computationally expensive because of the denominator term which increases with the size of the vocabulary which can be in hundred thousand range or even more. In order to tackle this issue , there are two methods of replacing the softmax function: 1) Hierarchical softmax . 2) Negative Sampling.

4.1.1 Hierarchical Softmax

Hierarchical softmax is implemented in the form of a binary tree with the vocabulary as the leaves. For a given $\mathbf{v_{w_I}}$, to get to the context word, a path of length $L\left(w\right)$ has to traversed starting from the root to the leaf of $\mathbf{v_{w_O}}$. Let $n\left(w,j\right)$ represent the j^{th} node of this path and $\mathrm{ch}\left(n\right)$ represent an arbitrary fixed child either to the right or left of the n. Finally let $[[x]]$ be 1 if x is True and -1 otherwise. Then,

$$p\left(w_O|w_I\right) = \prod_{j=1}^{L(w)-1} \sigma\left(\left[\left[n\left(w,j+1\right) = \mathrm{ch}\left(n\left(w,j\right)\right)\right]\right] \cdot {\mathbf{v_{n(w,j)}}}^\top \mathbf{v_{w_I}}\right) \tag{3}$$

Where $\sigma\left(x\right)$ is the sigmoid function, $v_{n(w,j)}$ is the representation (embedding) of the inner node encountered by $\mathbf{v_{w_I}}$ as it traverses to $\mathbf{v_{w_O}}$.
At each node $v_{n(w,j)}$, the dot product of v_{w_O} and that node is taken $(v_{n(w,j)}^\top v_{w_I})$, if the next node in the path is the right node , we consider $[[n\left(w,j+1\right) = \mathrm{ch}\left(n\left(w,j\right)\right)]]$ to be 1,

and if it's the left child, we consider it to be -1 (or vice versa, depending on the specific tree structure). This product

$$[[n(w, j+1) = \text{ch}(n(w, j))]] \cdot v_{n(w,j)}^{\top} v_{w_I}$$

is then passed to the sigmoid function , which determines its probability. The final probability $p(w_O|w_I)$ is obtained by multiplying all the probabilities along the path from the root to the node representing w_O.
The advantage of this method over the conventional softmax is that unlike the latter it doesn't have to evaluate the whole vocabulary W for the probability distribution , rather only $log_2(\mathbf{W})$.

4.1.2 Negative Sampling

Negative Sampling treats the output layer from a binary logistic regression perspective. Instead of dealing with the entire vocabulary W at once , it makes use of k negative samples for every positive sample.
There the problem gets reframed as the following: The output layer must output a 1 on the node that represents that context word associated with the center word and 0 other in other k nodes.

$$P(y = 1|w_t = w_I, w_{t+j} = w_O) = \sigma\left(\mathbf{v_{w_O}}^{\top}\mathbf{v_{w_I}}\right)$$

The loss function of this model is negative log likelihood of
$P(y = 1|w_t, w_O) \; + \; P(y = 0|w_t, w_O)$

$$= -log\left(\sigma\left(\mathbf{v_{w_O}}^{\top}\mathbf{v_{w_I}}\right)\right) + \sum_{i=1}^{k}\left[-log\left(1 - \sigma\left(\mathbf{v_{w_i}}^{\top}\mathbf{v_{w_I}}\right)\right)\right]$$

$$= -log\left(\sigma\left(\mathbf{v_{w_O}}^{\top}\mathbf{v_{w_I}}\right)\right) - \sum_{i=1}^{k} log\ \sigma\left(\left(-\mathbf{v_{w_i}}^{\top}\mathbf{v_{w_I}}\right)\right) \tag{4}$$

We need the loss to be maximum 1) when we have a positive example but the model predicts it to be negative , which can be achieved by a negative log function , this is the $-log\left(\sigma\left(\mathbf{v_{w_O}}^{\top}\mathbf{v_{w_I}}\right)\right)$ 2) when we have a negative sample and the model predicts it to be positive , this is the $-log\left(1 - \sigma\left(\mathbf{v_{w_i}}^{\top}\mathbf{v_{w_I}}\right)\right)$.
The authors experimentally showed that k can be in the range of 5-20 samples per iteration for a small dataset and as low as 2-5 samples per iteration for a large dataset. They also showed that negative sampling outperformed Hierarchical Softmax on the analogical reasoning task whereas Hierarchical Softmax with subsampling learns the best representation of words. Subsampling of frequent words aims to drop the most frequent words in a vocabulary like 'in' , 'the' , 'a' as they usually provide less information than a rare word. In order to achieve this systematically , we calculate the probability of a word w_i being dropped by the formula :

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}} \tag{5}$$

Where t is the threshold and $f(w_i)$ is the ratio between the number of times w_i appears and the total size of the vocabulary.

4.2 Section summary: Distributed Representation of words

In this section we the skip gram model and its output nodes. The skip gram is used to predict the center word given the context words. It can be used to train word embeddings with smaller corpora while CBOW is faster in training. We saw that the softmax function becomes computationally expensive to calculate as the size of the vocabulary increases. Therefore we can use hierarchal softmax or negative sampling.Hierarchal softmax is better in handling infrequent words as it uses the Huffman tree which is built considering the entropy of words and the negative sampling method outperforms the latter when in analogical reasoning tasks (Mikolov et al., 2013a)

5 The isomorphism Hypothesis

Word embeddings of two languages are said to be isomorphic if the translation of k nearest neighbors of a word x_w is the same as the k nearest neighbors of the translation of x_w. Figure 3 shows an example of two isomorphic graphs of words between English and Spanish

Figure 3: Representative isomorphic graphs between English and Spanish words relating to animals and food

Distributed word representation can capture a large amount of semantic information. In such representation , we are able to leverage such relationships using linear transformation. (Mikolov et al., 2013a) hypothesized that there exists such a linear relationship between the languages. The research showed that related words in one language and the translation of those words in another language has the same geometric representation in their embedding space. Thus they argued that the embedding spaces can be aligned by linear transformation as discussed in section 6.1. Unsupervised cross lingual word embedding are based on this assumption of isomorphism .

But (Søgaard et al., July 1, 2018) showed that this assumption is not true in general using Eigenvector similarity (section 5.1) and nearest neighbor graphs. They showed that the nearest neighbors in the monolingual word embedding space trained on (Conneau et al., 2017) for the top k English and German words nor nouns are the same despite the two languages being closely related to each other.

5.1 Eigenvector similarity

Although not all isospectral graphs are isomorphic , all isomorphic graphs are isospectral (Shigehalli and Shettar, 2011). (Søgaard et al., July 1, 2018) proposes the use of eigenvector similarity as a measure of isomorphism. Consider two languages L_1 and L_2 with word embeddings W_{L1} and W_{L2} that contains embedding of n words.

That is,

$$W_{L1} = \{w_1^{L1}, w_2^{L1}, w_3^{L1}, \ldots, w_n^{L1}\}$$
$$W_{L2} = \{w_1^{L2}, w_2^{L2}, w_3^{L2}, \ldots, w_n^{L2}\}$$

First , we build an adjacency matrix for these embeddings. An adjacency matrix of a language tells us to which words are each words in that language connected to.
The aim then is to see if the same words of languages are connected to the same set of words in both languages.

Consider the adjacency matrix of English words A_E and Spanish $A_S$from 3

$$A_E = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}, \quad A_{L1} \in \mathbb{R}^{n \times n}$$

$$A_S = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}, \quad A_{L1} \in \mathbb{R}^{n \times n}$$

The columns tells us to what words in English are the other words in English connected to. For example , the first column shows that 'cat' is connected to 'milk' and 'dog' (the words are numbered clockwise) . The same procedure goes for the adjacency matrix for Spanish
Next , both of these matrices needs to be normalized and the eigenvalues (λ) and associated eigenvectors ($\mathbf{v}$ of the normalized matrices needs to be found.
Then , we selected the smallest $\lambda \neq 0$ from each for each of the languages and the corresponding eigenvectors. In the case of our example :

$$\lambda_{E_{min}} = -1$$

$$v_{E_{min}} = [0.5, -0.5, 0.5, -0.5]$$
$$\lambda_{E_{min}} = -1$$

$$v_{S_{min}} = [0.5, -0.5, 0.5, -0.5]$$

Finally , we sort the eigenvectors in ascending order . The indices after sorting (indexed from 1) is as follow:

$$argsort\,(v_{E_{min}}) = [2, 4, 3, 1]$$

$$argsort\,(v_{E_{min}}) = [2, 4, 3, 1]$$

We infer the following mapping from our English vocabulary to Spanish vocabulary. node 2 corresponds to node 2 ; milk maps to leche
node 4 corresponds to node 4: dog maps to perro
node 3 corresponds to node 3: biscuit maps to galleta
node 1 corresponds to node 1: cat maps to gato

5.2 Hausdroff Distance

(Ding et al., 2024) uses Hausdroff distance to quantitatively measure isomorphism The algorithm is as follows:

1. For each $w_i^{L1} \in W_{L1}$, calculate the its cosine value to every element in W_{L2}. For each element keep the minimum value of their cosine angle with w_i^{L2} and take the maximum of the resultant values. ie, H(W_{L1})= $max(min(cos(w_i^{L1}, w^{L2}))$
2. Perform the same operation on w_i^{L2} : H(W_{L2})=$max(min(cos(w_i^{L2}, w^{L1}))$
3. Find the maximum between H(W_{L1}) and H(W_{L2}). ie , H= $max(H(W_{L1}), H(W_{L2}))$. H is called the Hausdorff distance. A higher Hausdorff distance denotes lower isomorphism.

6 Training methods

In this section we will discuss about some methods opted to train word embedding. We will primarily focus on 4 different methods:

1. mapping methods
2. unsupervised methods
3. joint methods
4. hybrid approaches

Mapping methods are based on the isomorphism hypothesis (section 5). (Mikolov et al., 2013a) proposed the use of a translation matrix that can be applied to the source word embedding to transform it to an embedding similar to that of the target. They used this technique on 4 pairs of language , with English being on of the languages in each pair. These pairs were En-Sp , Sp-En , En-Cz and Cz-En and obtained a P@1 score of 33% ,35% , 27% and 23% . These outperformed their baselines techniques such as edit distance , word co-occurrence count . The combination of translation matrix with edit distance performed even better with 43%, 44% , 29% and 23%. Their method was much less computationally expensive than the baselines. But the downside is that mapping methods require a large amount of parallel data. While this might be trivial for languages like English , it becomes a significant issue for low resource languages.
In order to alleviate this issue ,(Lample et al., 2017) proposed a framework that requires no parallel data to learn the translation matrix. The authors used adversarial training to

map both the source and target into the same latent space. (Conneau et al., 2017) also used adversarial training and a refinement step combined that outperformed supervised methods with their model reaching an accuracy score of over 66% compared to the 63.7% of supervised models. Note that here , supervised denotes the use of parallel data / bilingual signal for training.
(Søgaard et al., July 1, 2018) showed that the assumption of isomorphism doesn't hold equally for all language pairs , and differ especially for distant language pairs by measuring Eigenvector similarity. They also empirically showed that unsupervised learning models performed especially poorly for morphologically rich languages. The difference of performance was stark on English-Finnish and English-Estonian , for which the P@1 score less than 1%.
Joint training proposed by (Wang et al., 2019) combines both unsupervised and supervised techniques . This approach still required the use of large amount of parallel data. (Eder et al., August 1, 2021) built on this work but used anchor points which is a small set of bilingual signals to align the embeddings.(Hangya et al., 2023-01-01)(Woller et al., November 1, 2021)used related languages between the target and source which performed better than unsupervised for distant pair languages like English-Chuvash and English-Occitan. However, the performance is not consistent across all language pairs that were considered.
In this section we will delve into how these approaches work.

6.1 Mapping methods

6.1.1 Regression Method

Given there is an embedding x_s , we apply a linear translation matrix $W^{s \to t}$ such that $W^{s \to t} x_s$ finds a potential embedding of the translation. We learn the translational matrix using gradient descent by minimizing the Mean Squared Error(MSE) between the latter and the embedding of the translation of the corresponding word x^t.

$$\Omega_{MSE} = \sum_{i=1}^{n} \left|\left| W^{s \to t} x^s - x^t \right|\right|^2 \tag{6}$$

(6) can be rewritten as:

$$\Omega_{MSE} = \left|\left| W^{s \to t} X^s - X^t \right|\right|_F^2$$

We can solve for the translation matrix analytically using:

$$W^{s \to t} X^s = X^t$$

$$W^{s \to t} = X^+ X^t \tag{7}$$

Where X^+ is the Moore-Penrose pseudo inverse and is equal to $\left({X^s}^\top X^s \right)^{-1} {X^s}^\top$

6.1.2 Orthogonal methods

The analytical solution (Section 6.1) is computationally expensive; one way to solve it is using Singular Value Decomposition (SVD) under the constraint $W^\top W = I$. Take the SVD of ${X^t}^\top X^s$:

$$ {X^t}^\top X^s = U\Sigma V^\top $$

We place a constraint on $W^{s\to t}$ that it is orthogonal i.e, ${W^{s\to t}}^\top W^{s\to t} = I$ Thus :

$$ W^{s\to t} = UV^\top \tag{8} $$

Therefore, the $X_{aligned} = W^{s\to t}X^s$ Figure 4 shows the English-Dutch word embedding created through orthogonal mapping.

6.1.3 Canonical methods

The mapping is done using Canonical Correlation Analysis (CCA) (Ruan and Pircalabelu).
The model learns two translational matrices $W^{s\to}$and $W^{t\to}$ for source and target respectively and project them to a new joint space that's different from other of the two spaces.
We then find the correlation between the projected source language embedding with the projected target language embedding. This correlation is given by :

$$ \rho\left(W^{s\to}x^s, W^{t\to}X^t\right) = \frac{cov\left(W^{s\to}X^s, W^{t\to}X^t\right)}{\sqrt{var\left(W^{s\to}X^s\right) var\left(W^{t\to}X^t\right)}} \tag{9} $$

if $P = W^{s\to}X^s$ and $Q = W^{s\to}X^t$, We aim to find a_K and b_K that maximizes the correlation , equivalently , minimize the negative correlation .

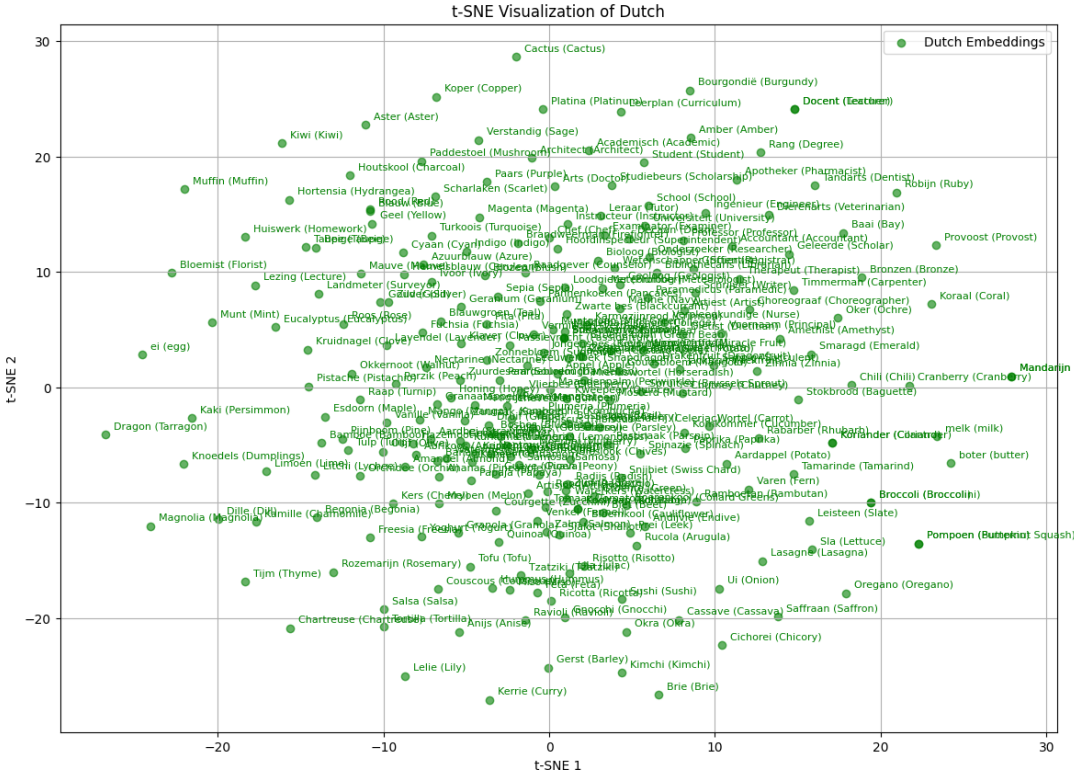
$$ \Omega_{CCA} = -\sum_{i=1}^{n} \rho\left(a_k^\top P, b_k^\top Q\right) \tag{10} $$

6.1.4 Margin methods

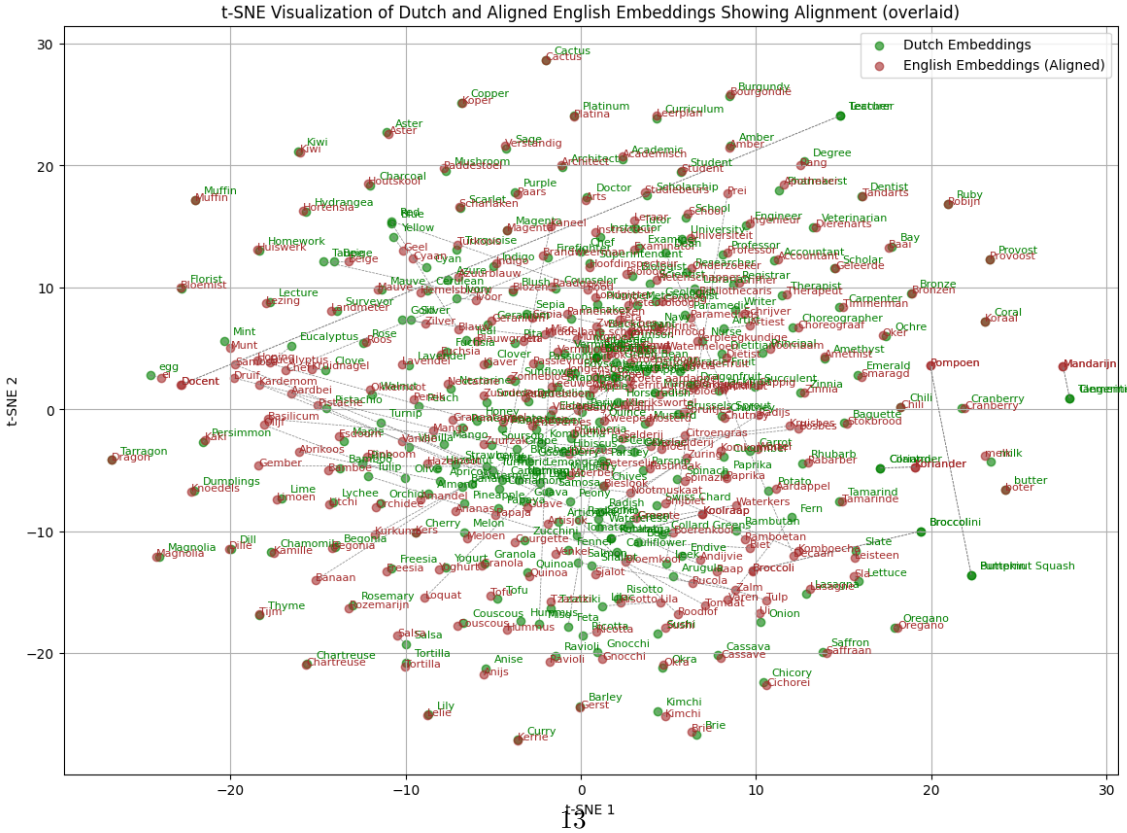
Mapping methods that uses MSE or ridge cost function can be affected by hubness 7.2.1, . In order overcome it , an alternative was proposed by (Lazaridou et al., July 1, 2015) . This approach considers the cosine similarity between the $W^{s\to t}x^s$ and x^t , $W^{s\to t}x^s$ and x^t_{neg} where , x^t_{neg} is a negative example from the target word embedding such that :

$$ \Omega_{MM} = \sum_{i=1}^{n}\sum_{j\neq i}^{k} max\{0, \gamma - cos\left(W^{s\to t}x_i^s, x_i^t\right) + cos\left(W^{s\to t}x_i^s, x^t_{neg_j}\right) \tag{11} $$

Where γ is a hyperparameter used for setting the margin between the correct and incorrect pairs. The authors were able to get a P@1 score of 33% compared to 28.7% by replacing the ridge cost function with max-margin.



(a) Target Dutch word embedding



(b) Dutch word embedding and aligned english word embedding overlaid.

Figure 4: Cross lingual English-Dutch word embedding using Orthogonal mapping. The grey lines shows words that remained unaligned

6.2 Unsupervised alignment using adversarial training

(Lample et al., 2017) proposes adversarial training to align word embeddings without parallel data. Consider the English word 'cat' and 'gato' for Spanish. These are from monolingual datasets $\mathcal{D}_{src}$ and $\mathcal{D}_{trgt}$. These translations are initially taken from a parallel dictionary learned from (Conneau et al., 2017). Each language has its own encoder $\mathcal{E}_{src}$ and $\mathcal{E}_{trgt}$. Similarly, they have their own decoders $\mathcal{D}_{src}$ and $\mathcal{D}_{trgt}$.
The source word passes through its encoder, the output of the encoder is latent space vector z^s_{cat} that is then passed through a discriminator that predicts the language of the vector as a binary output – 0 for source and 1 for target . The probability given by the encoder is $p\left(l|z^s_{cat}\right)$. Without losing generality, the probability is:

$$\prod_{j=1}^{m} p_D\left(l|z_j\right)$$

The discriminator is trained to predict the language. To do this, it minimizes the cross entropy loss :

$$\mathcal{L}_{\mathcal{D}}\left(\theta_D|\theta_E, \mathcal{Z}\right) = \ -E_{(x_i,l_i)} log\left[p_D\left(l_i|e\left(x_i, l_i\right)\right)\right] \tag{12}$$

Where θ_D and θ_E are the parameters of the discriminator and encoder respectively, $\mathcal{Z}$ are the encoder word embeddings and l represents the language.
This minimizes the loss when the discriminator predicts the correct language l_i given the encoder's output $e\left(x_i, l_i\right)$. That is the discriminator tries to predict '0' given the embedding of 'cat'. The encoder tries to fool the discriminator by minimizing:

$$\mathcal{L}\left(\theta_E, \mathcal{Z}|\theta_D\right) = \ -E_{(x_i,l_i)} log\left[p_D\left(l_j|e\left(x_i, l_i\right)\right)\right] \tag{13}$$

In our analogy, the encoder for English tries to fool the discriminator into predicting a 1 .To do this , the encoder tries to make the embedding of 'cat' to that of 'gato'. This is how the model aligns words and its translation into a shared embedding space.

6.3 Joint Training

Joint training refers to training the word embeddings simultaneously so that they benefit from shared words and other linguistic properties of languages. This method is especially useful in creating a bilingual word embedding between a high resource language and a low resource language.
(Wang et al., 2019) first creates a joint vocabulary for languages L_1 and $L2$. Then trains it using on the concatenated corpora using fastText. This however results in a coarse and ill-aligned corpora with additional problems such a oversharing – where a homophones of two languages that don't share the same meanings get embedded close to each other than words they are much more closely related to. For example: in French , *formation* means training which is not equivalent to the meaning of its English counterpart but these words might get embedded close to each other. In order to alleviate this issue , the authors proposes two additional steps : 1) Vocabulary Reallocation and 2) Alignment Refinement . For vocabulary relocation, they check the frequency of the shared words (words that appear in both languages) from V^s_j which contains the shared vocabulary and is a subset

of vocabulary V. For each token w in V_j^s , if the frequency of the token is more prominent in the corpus of a particular language, then that token is shifted to the subset of V that contains the words of that language only $V_j^{L_i}$. Else it remains in V_j^s. The used the following formula to calculate this ratio:

$$r = \frac{TL_2}{TL_1} = \frac{C_{L_1}(w)}{C_{L_2}(w)} \tag{14}$$

Where $C_{L_i}(w)$ is the count of w in language L_i and $T_{L_i}(w)$ is the total number of token in that language. The token w is allocated to the shared vocabulary if

$$\frac{1-\gamma}{\gamma} \leq r \leq \frac{\gamma}{1-\gamma}$$

For alignment refinement, any of the methods discussed in the previous sub-sections can be used whilst keeping the embedding of shared words untouched.

6.4 Hybrid approach : Mapping and joint training using related language

(Woller et al., November 1, 2021) extended (Wang et al., 2019) approach for creating a cross lingual word embedding between English and Occitan with an intermediary language. The authors provided the following reasons for this approach:

1. Unsupervised joint training can be affected by the embedding spaces of languages not being isomorphic and can create ill-aligned embeddings. Supervised joint training on the other hand requires a lot of parallel or comparable data which may not be available for low resource languages.

2. They proposed that using a related language could improve the quality of word embedding of the source language (here , Occitan) which can then help in improving the mapping to English.

The first step of their process is similar to the one proposed by (Wang et al., 2019) , with only a change in them using supervised MUSE (Conneau et al., 2017) for mapping instead of RCSLS (Alaux et al., 2018) For mapping shared embedding (Occitan-related Language) to English supervised MUSE was used .
One interesting result of their experiment was that despite their Occitan-English model outperforming the baseline in BLI P@1 tests, the English-Occitan model performed poorly . They attributed this to using French as the related language as English is French centric. However , the authors did not comment on why it performed poorly when the related language was Spanish. This could point to one flaw of using intermediary languages between Source and Target whereby if the Source is extremely centric to the related language , then the nearest neighbors of the language will mostly be the related language rather then the Target language.
(Sannigrahi and Read, 2022) attempts to combine both mapping and joint training to create a CLWE. For that , they used a related language in addition to the source and target language.
The related language is chosen such that it is topologically similar to the source language so that the isomorphism between the two word embeddings are preserved. Consequently,

mapping approach is used to align these word embeddings. Let resultant word embedding be X'.The target language was aligned to the related language using joint training. Let this embedding be Y'.

Since both X' and Y' are both aligned to the same related language , they are now isometric with each other. Therefore, they can be aligned with each other using the mapping approach. This approach is helpful in aligning distant pair languages that are not isometric to each other by first aligning them to a common language and then aligning the resultant embeddings to each other. This ensures that one can leverage the robustness of mapping approach for distant pair languages

6.5 Anchor based approaches

Mapping based approaches need a large amount of parallel text corpora which may not be feasible for moderate and low resource languages. (Eder et al., August 1, 2021) proposes an anchor based approach where the target language (typically a low resource language) is trained on top of a pre-trained word embedding of the source language using anchor points In order to achieve this , the monolingual word embedding (MWE) of the source language is created E_{L_1}. Then , we select some anchor points / seed words which are a set of words and their translations. For example, consider the case that we are interested in creating a shared word embedding for two languages L_1 and L_2. First the MWE of L_1 is created and then the representation of a seed word w_{iL_2} of L_2 in the target space is made to be the same as the representation of its translation w_{iL_1}. After aligning the seed words , the target space is then trained on non-seed words , that will now be aligned around the seed words creating the embedding E_{L_2}. The resultant bilingual embedding space is $E_{L_1L_2} = E_{L_1} \cup E_{L_2}$. This approach was shown to outperform traditional mapping approaches based on the acc@5 and acc@1 metrics. Another important advantage of this approach is that , it uses comparable data instead of parallel data for the embeddings which is much more practical than parallel data especially for moderate/low resource languages.

(Hangya et al., 2023-01-01) extended this approach using a chain of related languages between the the source language and the target language. They found that the anchor based approach directly from source to target is negatively affected as the 'distance' between the languages increases. In order to mitigate it , they used intermediate languages that are related to both source and target , mostly choosing intermediate languages that are in the same language family as the target language. For languages except Yakut (English – Russian – Yakut) and Swahili (English -German – Portuguese- Swahili) , their approach was shown to outperform the anchor based method , the traditional mapping methods and the unsupervised method which was the most competitive to their model in terms of performance . However they did not find any correlation between the amount of monolingual resource available and the target language their model performed the best. Despite better performance , the authors found the manual selection of intermediate languages to be a limiting factor . They also pondered on the effect of possible ordering of the languages in the chain. The authors noted that the quality of embedding spaces of intermediate language to be a pivotal factor in the performance of the model.

6.6 Section summary : Training methods

In this section we discussed supervised , unsupervised and semi-supervised training methods. Mapping methods fall under the supervised training method. They work by first collecting parallel corpora for source and target and then finding a translation matrix that aligns them. This can be done by regression method , by orthogonal mapping , canonical methods and margin methods. Orthogonal methods are much more computationally efficient comparing to the regression methods. Canonical methods are better adapted to polysemy (Ruan and Pircalabelu) and max-margin methods can be used instead of MSE by penalizing negative samples where the model considerably fails to approximate the target function. Mapping methods require a large amount of parallel data which makes it hard to be used on low resource languages.
Unsupervised methods don't use parallel data or bilingual signals and uses adversarial training to learn word embedding. However they have been shown to perform poorly on morphologically rich languages. Semi-supervised methods like joining training uses unsupervised method for initial training and mapping methods for refinement. Adding related languages as intermediate languages is shown to improve performance of these models. However , they perform poorly when the source or target language becomes too related language centric . Additionally , choosing the chain of related languages is a heuristic and complex task.

7 Challenges and Assumptions in Cross-Lingual Word Embeddings (CLWE)

7.1 Parallel data and low resource languages

Mapping approaches are by far one of the most robust methods of training word embeddings. However, as discussed in the previous sections they require large amount of parallel data and has limited work done on using comparable data severely limiting their application to low resource languages. With over 7000 languages in the world , there are only under 200 pre-trained word embeddings that is available for use (Facebook, 2016)(Costa-jussà et al., 2022) and most human annotated dataset is in English. Word2vec was trained on a billion words whereas languages like Kashmiri , Awadhi which have over 11 million native speakers combined don't have Wikipedia presence. Consequently, lacking quality word embeddings.

7.2 The curse of dimensionality

Word embeddings spans across multiple dimensions. In general, we take embedding to be a 300 dimensional vector as we have seen in section 4.1. Additionally , most of the work we have described here involves finding the nearest neighbors to an embedding. But there are some challenges in achieving this as we move higher up in dimensions. One of them is that , as we move up in dimensions the distance between n points to a point p becomes similar (Venkat, 2018) (Aggarwal et al., 2001). Consider an n-ball with radius 1. If we take 100 samples for each dimension 3,10,50,100 and 1000 and check the ratio of the points within 0.2 , 0,4 ... ,1 unit , we find that most of the points with dimensions 100 and above tend be in the same range of distance from the center of the sphere as shown in

the figure 5 . Consequently , the standard deviation of the cosine similarity between the points tends to decrease in higher dimensions as shown in figure .

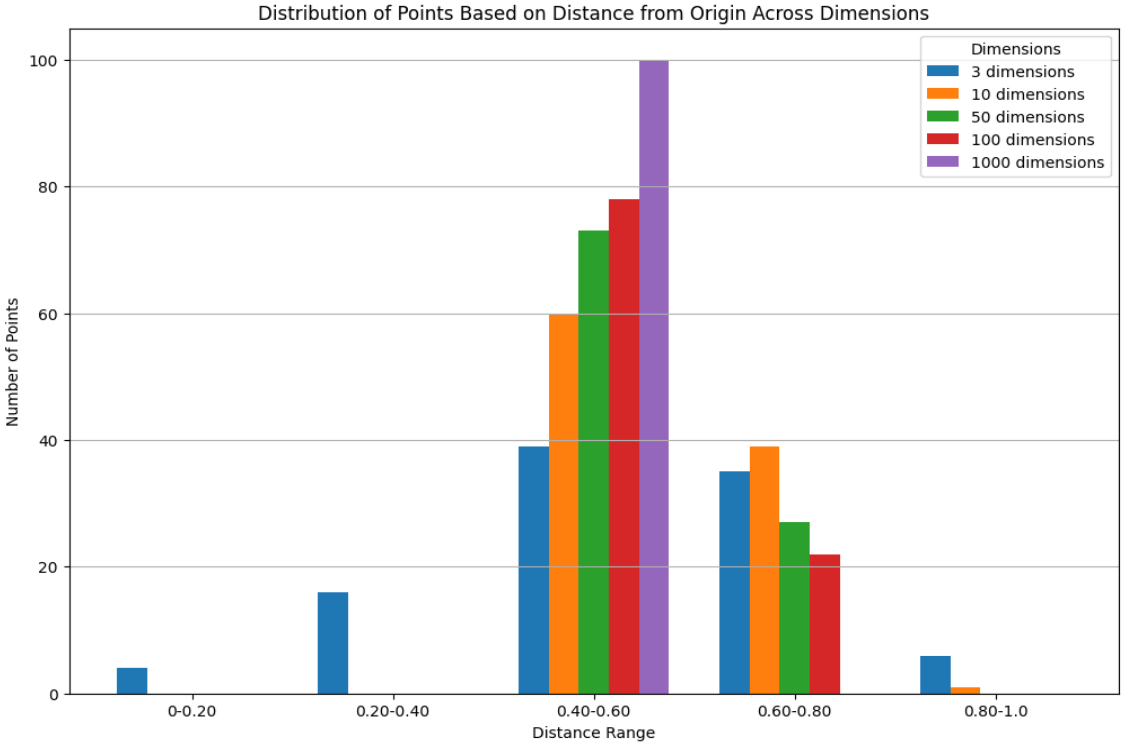


Figure 5: The data points with the lower dimension have a uniform spread in terms of their distance from the origin . As the dimensions increase , the data points tend to be with the same range of distance from each other

7.2.1 Hubness Problem

Hubs are points in high dimensional space that are 'nearest neighbors' to many other points without having any meaningful connection to them. This is a result of the distance concentration (Radovanovic et al., 2010). Hubness causes the points closer to the data-set mean to be closer to every other point and hence be included in their k nearest neighbors list without being meaningfully related to them thereby negatively impacting the results.This is exacerbated by metrics that ignore the relative distances between the points(Lazaridou et al., July 1, 2015).

Cross-Domain Similarity Local Scaling (CSLS) CSLS tries to mitigate the hubness problem by penalizing the similarity score when a vector lies in a dense area – areas with a large number of nearest neighbors (Ding et al., 2024).
For example , consider x^{L_1} as a word belonging to language L_1 , we check the cosine similarity of Wx^{L_1} (where W is the translation matrix) with its neighbors $N_T\left(Wx^{L_1}\right)$ in the language L_2 space . Similarly, we check the cosine similarity of the translation y^{L_2}

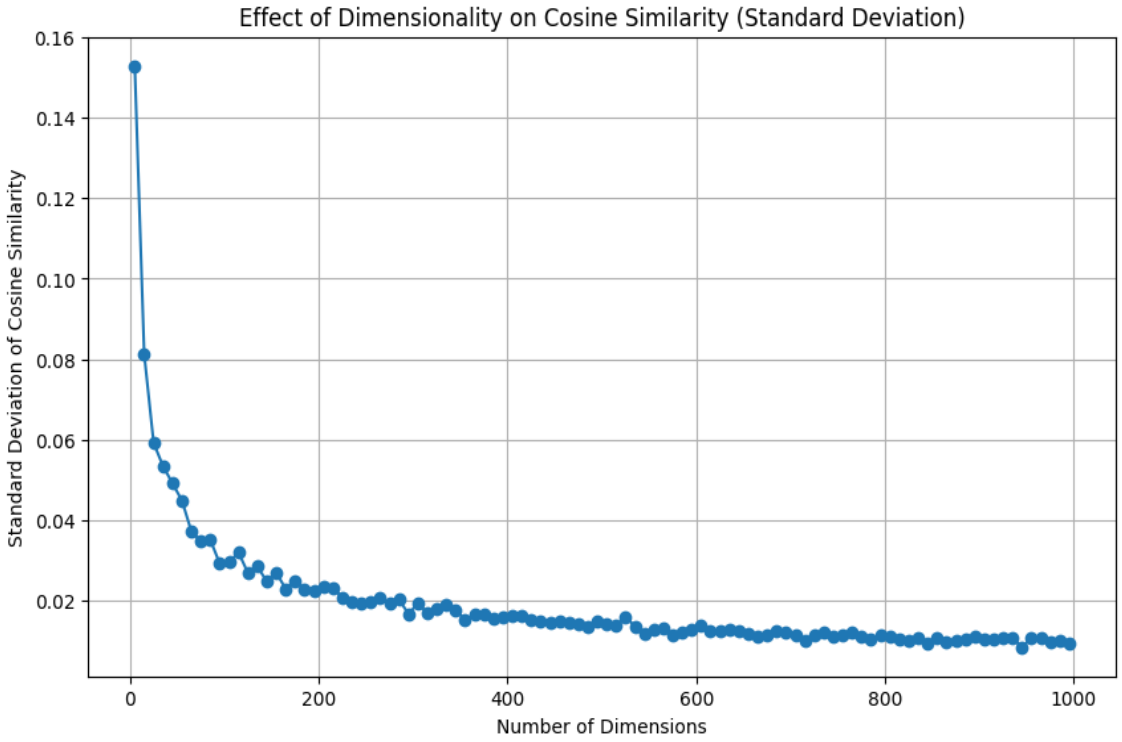


Figure 6: The figure shows the standard deviation of cosine similarity between 100 samples for dimensions 5 to 1000 with a step size of 10 . The standard deviation of cosine similarity decreases as dimensions increases

with its neighbors $N_S\left(y^{L_2}\right)$ in the L_1 space. The equation of CSLS is :

$$CSLS\left(Wx^{L_1}, y^{L_2}\right) = 2cos\left(Wx^{L_1}, y^{L_2}\right) - r_T\left(Wx^{L_1}\right) - r_S\left(y^{L_2}\right) \tag{15}$$

Where ,

$$r_T\left(Wx^{L_1}\right) = \frac{1}{K} \sum_{y \in N_T\left(Wx^{L_1}\right)} cos\left(Wx^{L_1}, y\right)$$

$$r_S\left(y^{L_2}\right) = \frac{1}{K} \sum_{x \in N_S\left(y^{L_2}\right)} cos\left(y^{L_2}, x\right)$$

r_T is measure of similarity the aligned source word has towards K elements in the target embedding , a high value here indicates presence of hubs. Similarly , r_S is the measure of similarity the target word has towards K elements in the source word embedding.

7.3 Impact of various modeling assumptions on the quality of CLWE

(Søgaard et al., July 1, 2018) showed the impact of various modeling assumptions-language similarity,domain difference,dimensionality and evaluation procedure- on the quality of CLWE which is summarized below:

7.3.1 impact of language similarity

They chose languages of different types - isolating , fusional , agglutinative and markings-dependent , mixed. Across all types and markings weakly supervised model outperformed unsupervised BLI. However the most interesting part of the experiment was the extremely poor performance of Unsupervised models for Estonian and Finnish ((P@1 $\times$ 100)=00.00 and 00.09 respectively) both of which are of mixed marking and agglutinative type.

7.3.2 impact of domain difference

The research used three datasets from different domains – Parliamentary Proceedings from EuroPal.V7 , EMEA Corpus and Wikipedia. The experiment showed that despite performing well on corpora from the same domain , the unsupervised methods failed in the performance when the corpora are from different domain. The minimally supervised method had a decent performance across multiple set ups. Both models performed poorly on different domain corpora for English-Finnish.

7.3.3 impact of dimensionality

Reducing dimensionality could downgrade the expressivity of a word embedding but it could also prevent overfitting. Therefore the authors used 40-dimensional (in contrast to the original 300 dimensional) word embeddings. The models performed worse for almost all languages except Hungarian , Finnish and Turkish. The authors hypothesized this to be due to the larger dimensional models overfitting to some peculiarities of the languages

7.3.4 impact of evaluation procedure

The authors analyzed the performance of the model based on different aspects of the query words: parts of speech , frequency bins , orthographic nature.
Parts of speech: Verbs showed the lowest performance compared to other parts of speech across all language pairs .
Frequency bins: The words were divided into groups based on their frequency- top 100 , 101st-1000th etc. The P@1 for EN-FI was zero across all groups.
Orthographic nature: The experiment showed that words that are a homographs to the translations but not homonyms performed worse than words that are neither homographs or homonyms.

7.4 Section summary: Challenges and Assumptions in Cross-Lingual Word Embeddings

In this section we discussed the lack of availability of parallel data and thereby quality embeddings for low resource languages. Then we discussed the curse of dimensionality and the hubness problem which causes a point to be a neighbor of several different points without being meaningfully connected to them and how it affects metrics like cosine similarity. We discussed how CSLS can be a useful measure in mitigating the effect of hubness by penalizing points in dense areas. Next, we discussed the impact of various modeling assumptions on the quality of word embedding. Unsupervised methods perform poorly on agglutinative languages with mixed marking . Whilst using comparable data

models performs worse if the data is from different domain. Low dimensional word embeddings seems to perform better for some morphologically rich languages like Finnish.

8 Evaluation

(Ruder et al., 2019) sorts different types of evaluations as 1) Intrinsic tasks 2) extrinsic tasks. Intrinsic tasks compares the word embeddings with human judgment datasets. These tasks often focuses on certain characteristics of words like their similarity , relatedness ,association, Part Of Speech (POS) etc in a controlled environment.
Word Similarity task checks how well the vector space representation of words matches human understanding of word similarity. Multilingual word similarity datasets are extensions of those used to evaluate English word embeddings, adapted for multiple languages.
QVEC (Tsvetkov et al., 2015) is a method used to measure how well word embeddings capture linguistic information by maximizing their correlation with a human annotated dataset by performing CCA. Multi QVEC+ is an extension of it available in multiple languages.
Extrinsic tasks on the other hand , focuses much more on the applicability to downstream tasks . Extrinsic tasks includes word alignment and Bilingual Language Induction (BLI). In word alignment prediction, each word in a sentence from the source language is matched to the most similar word in the target language sentence. If a source language word is not in the vocabulary, it isn't aligned to any word. Target language words that are out of vocabulary get the lowest similarity score and are not matched to any source language word.
BLI (Glavas et al., 2019) is the most commonly used method for evaluating cross lingual word embeddings by measuring the probability of the target word being in the k nearest neighbor of the translated source word. In the subsequent subsections we will take a

deeper look into measuring word similarity and BLI.

8.1 Measuring word similarity

Similar words are those that have semantic similarity , this is directly contrasted with associated words which could co-occur in a given corpora but may not necessarily have any commonality physically , functionally or categorically. For example , *coffee* and *juice* are more semantically similar than *coffee* and *cup*.
There are a number of human annotated datasets that describes the relatedness / similarity of word pairs. These datasets are then compared against the word vectors from word embeddings like word2vec (Mikolov et al., 2013b).
Some well known datasets are:

1. RG dataset
2. MC dataset
3. WordSim-353 dataset
4. SimLex-999

5. Multi-Simlex

Figure 7 shows the monotonic relationship between Simlex-999 (Hill et al., 2015) and word2vec embeddings. The spearman's rank coefficient for the same is 0.44.

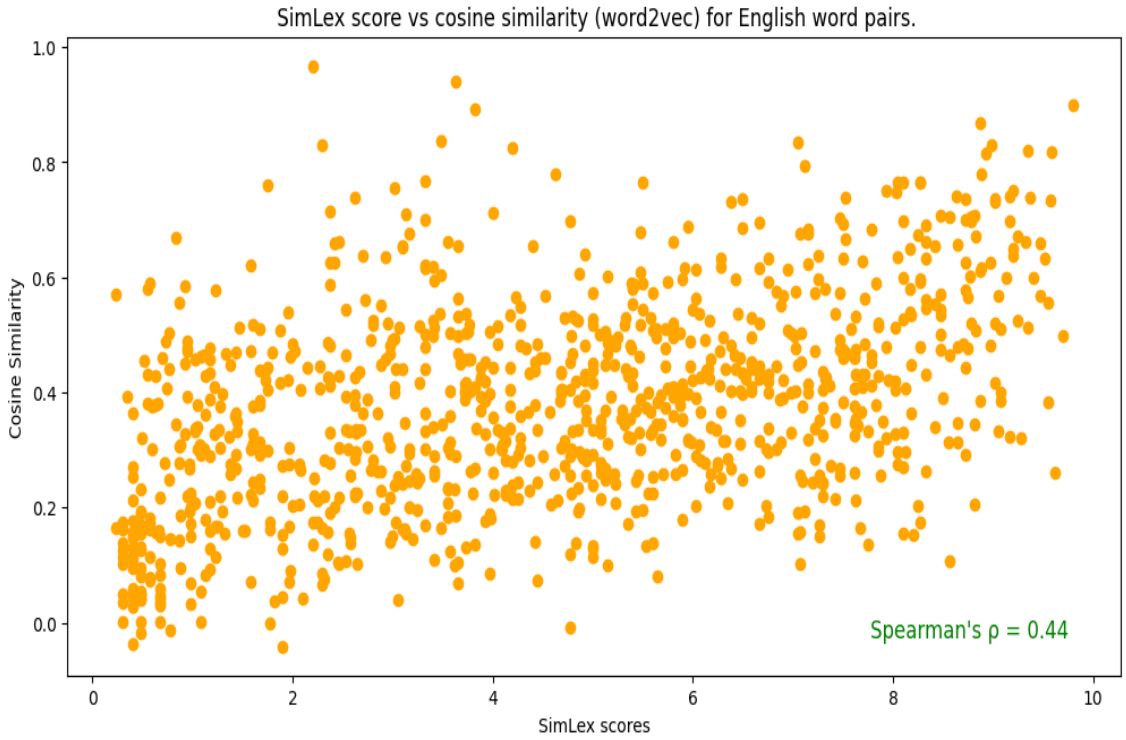


Figure 7: The figure shows monotonic trend between the scores of word pairs in the SimLex dataset and the value of cosine similarity of their vectors in word2vec. The spearman's rank correlation coefficient was 0.44 indicating a weak monotonic relationship.

(Vulić et al., 2020) extended SimLex to 12 typologically diverse languages. The mean scores given to word pairs by the annotators in English and Arabic have a strong positive monotonic relationship as shown in Figure 8 making it useful in the evaluation of cross lingual word embeddings.

8.2 Bilingual Lexicon Induction (BLI)

Given a source-target word pair (w^s, w^t) derived from the cross lingual embedding space , the task of BLI is to find the percent of occurrence of the target word as the k^{th} nearest neighbor. This measure is the $P@k$. For example a $P@1$ score of 15 % would mean for 15% of the queries , $w^t$was also the closest to the source word in the standard dictionary (analogous to setting k as 1 in knn). MUSE (Conneau et al., 2017) is the baseline for most researches. $P@1$ is considered to be a rigid measure , so $P@5$ and $P@10$ are also in conjunction.

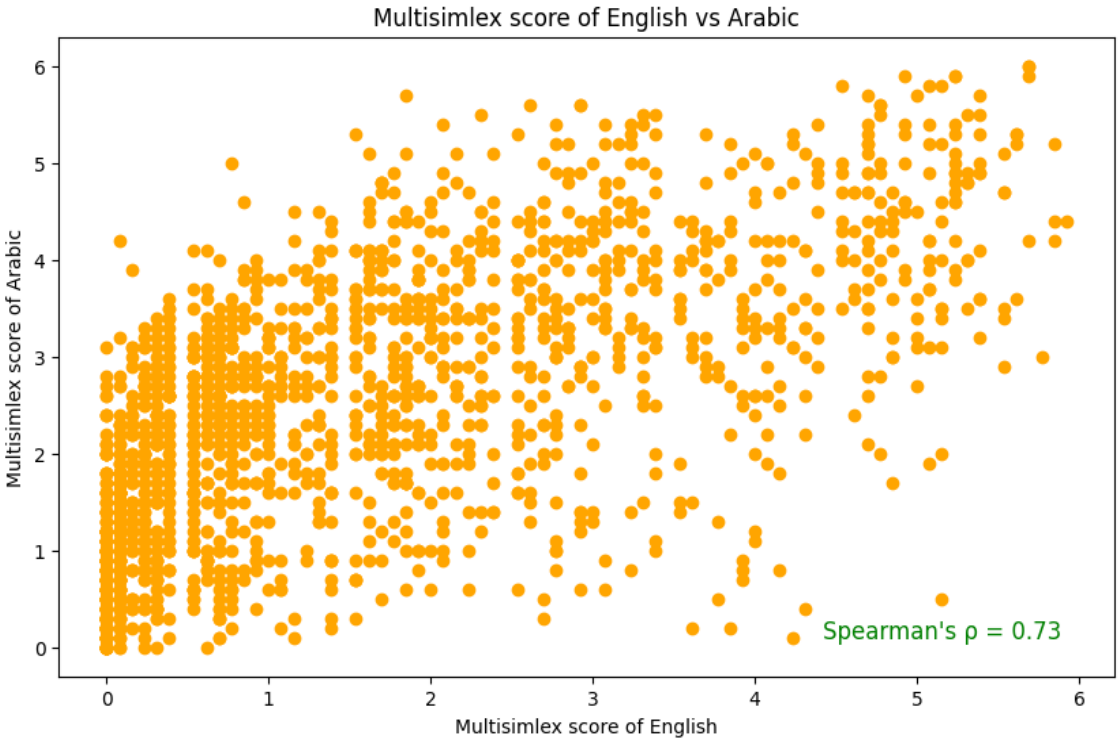


Figure 8: The figure shows monotonic trend between the scores of English and Arabic word pairs in the Multi-SimLex dataset. The spearman's rank correlation coefficient was 0.73 indicating a strong monotonic relationship.

8.3 Section summary: Evaluation

In this section we explored different evaluation tasks, both intrinsic and extrinsic. Human annotated datasets play a major role in differentiating similar words from associated words. Word embedding models are evaluated against these dataset to measure their performance in terms of capturing semantic relationships. Bilingual lexicon induction is the task of automatically creating a dictionary between two languages. It involves finding word pairs that have the same meaning in both languages, usually by comparing their vector representations in word embeddings. This process helps align words from one language with their equivalents in another, which is useful for machine translation and other cross-lingual tasks. P@K refers to the probability of the target word being in the k nearest neighbors. P@1 is the most commonly used measure with P@5 and P@10 used in conjunction due to its rigidity.

9 Visualizing word embeddings using TSNE[t-distributed stochastic neighbor embedding]

T-SNE is a dimensionality reduction technique that plots a high dimensional data into a two dimensional or three dimensional space for visualization. Figure 4 uses T-SNE for visualization. There are three main steps to T-SNE :

1. Constructing Probabilities in High Dimensions
2. Constructing Probabilities in Low Dimensions
3. Minimizing the Kullback-Leibler (KL) Divergence

9.0.1 Constructing Probabilities in High Dimensions

Consider a point x_i , let x_j represent the neighbor in its cluster. We have to first find the Euclidean distance between x_i and these neighbors.

$$dist(x_i, x_j) = ||x_i - x_j||^2$$

The similarity between the these two points is then defined by a Gaussian distribution centered at x_i. The standard deviation of distribution is dictated by the *perplexity* which is based on 'tight' or 'loose' the groups are (Figure 9). We then calculate $P_{j|i}$ which the

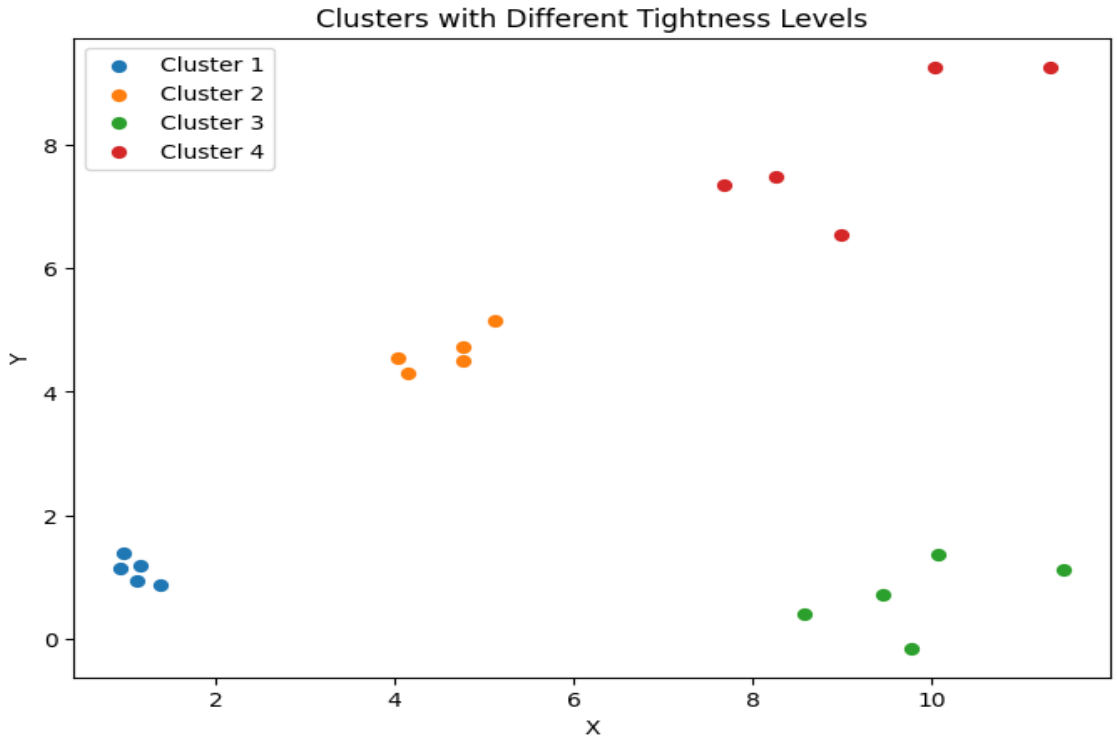


Figure 9: Clusters with different level of 'tightness'. Cluster 1 is the most closely packed hence it will have the lowest perplexity (lowest variance) compared to other clusters. Cluster 4 will have the highest perplexity

conditional probability of x_j being the neighbor of x_i and is given by:

$$P_{j|i} = \frac{\frac{exp(-||x_i - x_j||^2)}{2\sigma_i^2}}{\sum_{k \neq i} \frac{exp(-||x_i - x_k||^2)}{2\sigma_i^2}} \tag{16}$$

The reason for the division by summation is because, depending on the group they belong , x_i will have different distributions. In order to normalize these differences and bring the values to be in the range between 0 and 1 , we divide by the summation. Due to the differences in distribution , $P_{i|j}$ and $P_{j|i}$ are not symmetric. Therefore, we symmetrize the probability distribution by taking the average of $P_{i|j}$ and $P_{j|i}$ over all data points as shown

$$P_{ij} = \frac{P_{i|j} + P_{j|i}}{2N}$$

Where N is the number of all data points and $P(i, j)$ is the joint probability of x_i and x_j. This is an undirected measure of "closeness" of these points agnostic of their distributions.

Perplexity: Perplexity is a user-defined measure of how many points should be considered as a nearest neighbors of a point. Figure 4 has perplexity set to 20. This in turn dictates the variance of the Gaussian distribution. If we need a high perplexity we need to have high variance and vice versa. The perplexity P of x_i is given by:

$$P = 2^{H(P_{j|i})}$$

where, $H(P_{j|i})$is the Shannon entropy and is given by $-\sum_j P_{j|i} log_2 P_{j|i}$ Finding σ_i for a given perplexity is an iterative task with the following steps :

1. Calculate the targeted entropy based on the given perplexity.
2. Calculate the entropy $H(P_{j|i})$ based on a randomly initialized σ_i
3. Adjust σ_i until the entropy $H(P_{j|i})$ produces a perplexity close to the target.

9.0.2 Constructing Probabilities in Low Dimensions

Let y_i represent the the position of x_i in low dimension. Its value will be adjusted iteratively to capture the structure in the original high-dimensional space. First , the points are randomly initialized. Then we calculate the conditional probability $q_{(j|i)}$ which is calculated from the Student's t-distribution (with $df = 1$) as shown:

$$q_{(j|i)} = \frac{\left(1 + ||y_j - y_i||^2\right)^{-1}}{\sum_{k \neq i} \left(1 + ||y_j - y_i||^2\right)^{-1}} \tag{17}$$

The goal is to move the y_i points around in the low-dimensional space until the probability distribution $q_{(ji)}$ is as close to the value of $p_{(ji)}$. In order to achieve this, we will minimize the KL divergence of these probabilities by performing gradient descent. The equation of KL divergence of the two distributions P and Q is :

$$KL(P||Q) = \sum_{i \neq j} p_{(ji)} log \frac{p_{(ji)}}{q_{(ji)}} \tag{18}$$

The derivative represents the cost function of T-SNE .

$$\frac{\partial KL}{\partial y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j) \left(1 + ||y_i - y_j||^2\right)^{-1}$$

If $p_{ij} > q_{ij}$, y_i and y_j are pulled closer. Else, its pushed apart. The algorithm repeatedly updates each y_i in the direction that minimizes KL Divergence until p_{ij} approximately q_{ij}

Why Student's t-distribution? Unlike the Gaussian used in the high-dimensional space, the t-distribution has heavy tails. This means that it allows for larger distances between points, helping to prevent the "crowding problem" (Van der Maaten and Hinton, 2008), where points all collapse into a small area in the low-dimensional space (Figure 10).

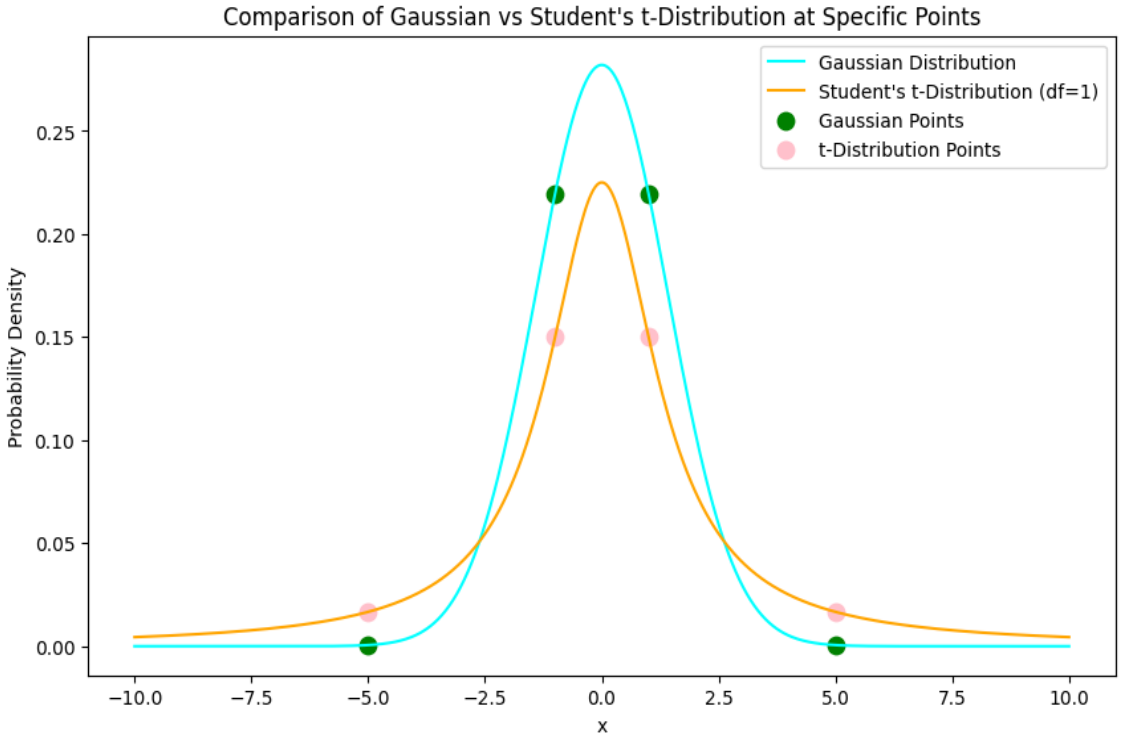


Figure 10: The Gaussian distribution probabilities and t-distribution probabilities of the same four points. We can see that for the same value , t-distribution has a slightly higher probability than the Gaussian. This will be helpful in giving more granularity in mapping points that are farther apart from x_i

10 Open problems and Future directions

Apart from the challenges discussed in previous section , there are some open problems and my suggestions that I would like to discuss in this section . Firstly , parallel data is one key issue that we have discussed in the previous sections. There needs to be unified effort like that of (Facebook, 2016) (Costa-jussà et al., 2022) (Hill et al., 2015) to gather parallel data. However, this is definitely challenging and we need to check for alternative methods as well.

Languages across the world shares loan words. This is true also for low resource languages. The efficacy of using loan words as bilingual signals could be evaluated. To illustrate this idea , consider this sentence in Spanish: "El restaurante tiene un menú con sushi y ramen muy popular entre los clientes." A non-Spanish speaker can infer the meaning of this sentence. If we use a shared embedding space for loan words and train a skip-gram model

to predict other words in the sentence, the model could align English words with their equivalents in the target language—for example, aligning 'muy' with 'very'.
Since movies in streaming platforms tend to have titles in several different languages and dialects , if part of older movie/series subtitles were made open source , it would serve help to create a large collection of parallel/comparable data. Of course , one has to respect the IP right of the creators of the movie.
Although using anchor points to align languages is highly promising, one must be careful not to force one language's embeddings to conform too closely to the other, as this can cause the loss of its unique characteristics. For instance, what is considered food can vary across cultures, so it's important to ensure that these differences in concepts and categories are preserved in the resulting word embeddings
There seems to be no method that performs equally across all languages , for example Unsupervised methods outperforms Hangya et al. (2023-01-01) in English-Swahili whereas the latter performs better for English-Finnish. Efforts could be made to understand the reason behind it with a domain expert.
Efficacy of using lower dimensional word embeddings to represent agglutinative/fusional languages (Søgaard et al., July 1, 2018) could be explored further.
'Simpler' languages could be aligned to languages that have more expressive efficiency than the opposite so that the language doesn't lose its efficiency.
If a language/dialects have several cognates /loan words , then using a chain of related languages might be effective because it enables the embedding to capture these characteristics of the language effectively. This can be further useful in dialects.
Its important that we understand the nature of the language that we are working with in order to effectively use the right methods. Its form , type, dependency marking all could play a vital role in how a method will perform on it.

Acknowledgments and Disclosure of Funding

I would like to thank Dr.Mohamed Alhajri for giving me an opportunity to do this survey. I would like to thank all the YouTube , Medium and Wikipedia and other independent creators and editors for creating materials of value that was helpful to me throughout the survey.

References

C. C. Aggarwal, A. Hinneburg, and D. A. Keim. On the surprising behavior of distance metrics in high dimensional space. In *Database theory—ICDT 2001: 8th international conference London, UK, January 4–6, 2001 proceedings 8*, pages 420–434. Springer, 2001.

J. Alaux, E. Grave, M. Cuturi, and A. Joulin. *Unsupervised Hyperalignment for Multilingual Word Embeddings*. -11-02 2018. URL https://www.researchgate.net/publication/328758621_Unsupervised_Hyperalignment_for_Multilingual_Word_Embeddings.

D. Bollegala, R. Kiryo, K. Tsujino, and H. Yukawa. Language-independent tokenisation rivals language-specific tokenisation for word similarity prediction. *arXiv preprint arXiv:2002.11004*, 2020.

V. Boykis. What are embeddings. *10.5281/zenodo*, 8015029, 2023.

A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. J'egou. Word translation without parallel data. *ArXiv*, October 11, 2017. URL https://www.semanticscholar.org/paper/Word-Translation-Without-Parallel-Data-Conneau-Lample/562c09c112df56c5696c010d90a815d6018a86c8.

M. R. Costa-jussà, J. Cross, O. Çelebi, M. Elbayad, K. Heafield, K. Heffernan, E. Kalbassi, J. Lam, D. Licht, J. Maillard, et al. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*, 2022.

Q. Ding, H. Cao, Z. Feng, and T. Zhao. Enhancing isomorphism between word embedding spaces for distant languages bilingual lexicon induction. *Neural Computing and Applications*, 36(24):15091–15102, -08-01 2024. doi: 10.1007/s00521-024-09837-1. URL https://doi.org/10.1007/s00521-024-09837-1.

T. Eder, V. Hangya, and A. Fraser. Anchor-based bilingual word embeddings for low-resource languages. page 227–232, Online, August 1, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-short.30. URL https://aclanthology.org/2021.acl-short.30.

I. Facebook. *fastText: Library for fast text representation and classification*, 2016. URL https://github.com/facebookresearch/fastText.

G. Glavas, R. Litschko, S. Ruder, and I. Vulic. How to (properly) evaluate cross-lingual word embeddings: On strong baselines, comparative analyses, and some misconceptions. *arXiv preprint arXiv:1902.00508*, 2019.

V. Hangya, S. Severini, R. Ralev, A. Fraser, and H. Schütze. Multilingual word embeddings for low-resource languages using anchors and a chain of related languages. pages 95–105, 2023-01-01. doi: 10.18653/v1/2023.mrl-1.8. URL https://www.researchgate.net/publication/376405172_Multilingual_Word_Embeddings_for_Low-Resource_Languages_using_Anchors_and_a_Chain_of_Related_Languages.

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

G. Lample, L. Denoyer, and M. Ranzato. Unsupervised machine translation using monolingual corpora only. *ArXiv*, October 31, 2017. URL https://www.semanticscholar.org/paper/Unsupervised-Machine-Translation-Using-Monolingual-Lample-Denoyer/e3d772986d176057aca2f5e3eb783da53b559134.

A. Lazaridou, G. Dinu, and M. Baroni. Hubness and pollution: Delving into cross-space mapping for zero-shot learning. page 270–280, Beijing, China, July 1, 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1027. URL https://aclanthology.org/P15-1027.

T. Mikolov, Q. V. Le, and I. Sutskever. Exploiting similarities among languages for machine translation. *ArXiv*, September 16, 2013a. URL https://www.semanticscholar.org/paper/Exploiting-Similarities-among-Languages-for-Machine-Mikolov-Le/0157dcd6122c20b5afc359a799b2043453471f7f.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. volume 26. Curran Associates, Inc., 2013b. URL https://papers.nips.cc/paper_files/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html.

M. Radovanovic, A. Nanopoulos, and M. Ivanovic. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11(sept): 2487–2531, 2010.

W. Ruan and E. Pircalabelu. Word embeddings using canonical correlation analysis.

S. Ruder, I. Vulić, and A. Søgaard. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research*, 65:569–631, 2019.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, October 1, 1986. doi: 10.1038/323533a0. URL https://www.nature.com/articles/323533a0.

S. Sannigrahi and J. Read. *Isomorphic Cross-lingual Embeddings for Low-Resource Languages*. -03-28 2022. URL https://www.researchgate.net/publication/359517508_Isomorphic_Cross-lingual_Embeddings_for_Low-Resource_Languages.

V. S. Shigehalli and V. M. Shettar. Spectral techniques using normalized adjacency matrices for graph matching. *Int. J. Comput. Sci. Math*, 2(4):371–378, 2011.

A. Søgaard, S. Ruder, and I. Vulić. On the limitations of unsupervised bilingual dictionary induction. page 778–788, Melbourne, Australia, July 1, 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1072. URL https://aclanthology.org/P18-1072.

Y. Tsvetkov, M. Faruqui, W. Ling, G. Lample, and C. Dyer. Evaluation of word vector representations by subspace alignment. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2049–2054, 2015.

L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

N. Venkat. The curse of dimensionality: Inside out. *Pilani (IN): Birla Institute of Technology and Science, Pilani, Department of Computer Science and Information Systems*, 10, 2018.

I. Vulić, S. Baker, E. M. Ponti, U. Petti, I. Leviant, K. Wing, O. Majewska, E. Bar, M. Malone, T. Poibeau, et al. Multi-simlex: A large-scale evaluation of multilingual and crosslingual lexical semantic similarity. *Computational Linguistics*, 46(4):847–897, 2020.

Z. Wang, J. Xie, R. Xu, Y. Yang, G. Neubig, and J. Carbonell. Cross-lingual alignment vs joint training: A comparative study and a simple unified framework. *ArXiv*, October 10, 2019. URL `https://www.semanticscholar.org/paper/Cross-lingual-Alignment-vs-Joint-Training%3A-A-Study-Wang-Xie/5884948777dfc003ba49e1513420830616281839`.

L. Woller, V. Hangya, and A. Fraser. Do not neglect related languages: The case of low-resource occitan cross-lingual word embeddings. page 41–50, Punta Cana, Dominican Republic, November 1, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.mrl-1.4. URL `https://aclanthology.org/2021.mrl-1.4`.