Computational Linguistics - 2

Evaluating Performance of Various Word Embedding Techniques across Different NLP Tasks

Submitted By: Ishaan Romil (2023114011)
Submitted To: Dr. Parameswari Krishnamurthy, Dr. Rajakrishnan Rajkumar
Date of Submission: 4th December, 2024

Acknowledgement

I express my sincere gratitude to Parameswari ma'am and Rajakrishnan sir for their invaluable teachings and insights throughout the past semester, which have been instrumental in shaping this report.

Additionally, I extend my thanks to Uday sir and Ketaki ma'am for their unwavering assistance in navigating doubts and difficulties, contributing significantly to my enhanced understanding of computational linguistics in general, and the application of word embedding techniques across various downstream natural language processing applications in particular.

Special appreciation goes to my friends and family for their steadfast support.

Abstract

Word embeddings in natural language processing (NLP) refer to the representation of words for text analysis in the form of real-valued vectors. These embeddings capture the semantic meaning of words by positioning similar words close to each other in this space. This paper presents a comprehensive evaluation of the performance of three popular word embedding techniques – Word2Vec, GloVe, and FastText – across a variety of tasks across the NLP space. Namely, the tasks this paper will focus on include named-entity recognition, sentiment analysis, and POS tagging, utilizing datasets in both English and French. The performance of these techniques on these tasks will give us a deeper understanding of the efficacy of the techniques across various use cases, allowing us to gain insights into the potential advantages and drawbacks of each model.

Before implementing the techniques themselves, I will make a prediction about which technique will have the highest accuracy across the aforementioned tasks and which technique will be the most consistent across generalized NLP applications. I will also make separate predictions with respect to the efficacy of these techniques across languages. For the evaluation, I will train models using publicly available datasets in both languages and compare performance metrics such as precision, recall, and F1 score, alongside a more detailed comparison via the use of a confusion matrix. These metrics will allow us to make a final conclusion regarding the hypothesis. It will also allow us to take note of phenomena that pervade within and between the various embedding techniques.

Index

Introduction

Research Question

"Which word embedding is best suited for a general-purpose, multi-language NLP application?

Word embeddings, as foundational tools in natural language processing, are manifestations of how to represent textual data in a computationally processible fashion. They map words or phrases to dense, continuous vector spaces, capturing semantic and syntactic relationships that enable downstream NLP tasks like named entity recognition (NER), sentiment analysis, and part-of-speech (POS) tagging. Among these, three embeddings are noteworthy: Word2Vec, GloVe, and FastText.

However successful as they may be in many applications, the strength of those embedding techniques can significantly differ based on the task undertaken and the linguistic context of the inputs. While Word2Vec excels at capturing local syntactic relationships through predictive modeling, GloVe's approach is more suited to encoding global semantic associations based on co-occurrence. In contrast, FastText introduces subword-level representation, making it better suited for handling morphologically rich languages and OOV words.

The research question is highly relevant in the current era of globalization, where NLP systems are often used across several languages and domains. While English is offered with relatively simple morphology and abundant linguistic resources that most embedding techniques thrive on, morphologically rich languages such as French introduce complex inflectional structures and very sparse word representations, which could expose some limitations of traditional word embeddings. In addition, varying NLP tasks put different requirements on the embedding, which makes the pursuit of an approach universally optimal even more challenging.

This paper conducts a systematic comparison between Word2Vec, GloVe, and FastText across three well-known NLP tasks—namely, named entity recognition, sentiment analysis, and POS tagging—on both English and French datasets. The research compares, then, the task- and language-specifics of the embedding in order to pinpoint which technique has consistency as well as effectiveness for general-purpose multilingual NLP applications. This way, it will clarify the comparative strengths and weaknesses of these techniques, besides shedding light on their adaptability across linguistics and functional contexts.

Literature Review

Word2Vec

Word2Vec was first introduced by Mikolov et al. in 2013. It is a neural network-based model that generates dense, real-valued representations of words by exploring the relationships among words. The resulting embeddings represent words in a high-dimensional continuous vector space where words with similar meanings are placed close to each other, indicating their semantic and syntactic relationships. This model has become an important tool in NLP, due to its ability to capture linguistic information efficiently.

Two major configurations exist for Word2Vec. These are Continuous Bag of Words (CBOW) and Skip-Gram. CBOW predicts the target word from the surrounding context words. For example, in the sentence fragment "The cat is on the," CBOW could predict the word "mat." This framework has computational efficiency and works very well with large datasets. The Skip-Gram model, in contrast, predicts the context words from a given target word. For instance, using the word "cat," the model predicts words that would likely follow it in a sentence: "The" and "is." Although computationally more expensive than CBOW, Skip-Gram works better when the dataset is small or if there is diversity in linguistic features.

In order to make the training process computationally feasible, Word2Vec uses optimization techniques such as negative sampling and hierarchical softmax. Negative sampling reduces the cost of the large vocabulary by updating only a limited number of weights corresponding to randomly selected non-context words. Hierarchical softmax accelerates speed by making the vocabulary into a binary tree where the probabilities of words can be calculated in time.

A prominent merit of Word2Vec embeddings is that they can potentially summarize semantics and syntax between the words. For example, through vector arithmetic operations of those embeddings, one ends up with meaningful results: Hitler − Germany + Italy ≈ Mussolini.
This ability stems from the fact that the model is encoding word relationships as linear translations in vector space.

Word2Vec has been used for a variety of downstream natural language processing tasks, including sentiment analysis, text classification, and machine translation. This is due to its efficacy in processing large datasets and capacity to deliver embeddings that are very semantically rich. However, Word2Vec is not free from disadvantages. A major drawback of this method is its inability to incorporate contextual information since each word is assigned a single embedding regardless of its connotation in different contexts. For example, the word "bank" would have the same vector representation regardless of whether it refers to a financial institution or the bank of a river. In addition, Word2Vec suffers from out-of-vocabulary words,

since it cannot generate embeddings for words that were not encountered during training. This limitation makes it less effective in resource-poor settings or in linguistically morphologically complex languages.

The empirical testing of Word2Vec reflects its high utility in different applications. Baroni et al. (2014) compared Word2Vec with other embedding models against word similarity and analogy tasks, showing that it could capture semantic relationships. It is less ideal for use in morphologically rich languages or OOV-intensive environments because of its static embeddings and lack of subword representation. In a multilingual environment, FastText, which captures the information of subwords, is often better than Word2Vec.

Word2Vec remains one of the foundational models in NLP, a compromise between efficiency and semantic richness. Although its shortcomings spurred more advanced techniques, in the context of available large well-defined datasets, when efficiency is a top priority, Word2Vec continues to be a great resource.

GloVe

GloVe (Global Vectors for Word Representation) was developed by Pennington, Socher, and Manning in 2014 to create word embeddings utilizing global statistical information. Unlike Word2Vec, which focuses on context windows, GloVe devises embeddings based on statistics about co-occurrences of words across the corpus and is therefore quite notably skilled at identifying more contextual relationships between words.

The co-occurrence matrix forms the core of GloVe, which records the frequency at which words co-appear within a corpus. For example, in a matrix derived from a textual corpus, the entry located at the row i and column j may be the number of times the word i co-occurs with word j.
It creates a matrix that reflects relative likelihoods of word co-occurrences within the context window. This way, it presents an approach that is analytic in character and not predictive, unlike the paradigm of Word2Vec and other such models.

The core essence of GloVe lies in the objective function, which emphasizes that the word co-occurrence has ratios of probabilities. The objective for GloVe is to obtain word embeddings based on the following relationship: the dot product of two word vectors should be close to the logarithm of their co-occurrence probability. This principle is based on linguistic phenomena such as transitivity in analogies. To illustrate, consider the analogy Hitler − Germany + Italy = Mussolini, the embedding for "Mussolini" should maintain proportional relationships with other words in the space.

To create embeddings, GloVe uses a weighted least squares cost function that is optimized to maximize the importance of both frequent and infrequent co-occurrences. The function also decreases the noise in the training data set by giving reduced weight to very rare or very frequent word pairs (Pennington et al., 2014). Additionally, GloVe improves its performance by using a truncated co-occurrence matrix, which greatly reduces the computational overhead involved in processing large vocabularies.

The other advantage with GloVe is its ability to discern both direct and indirect associations amongst terms. In this case, ‘direct association’ denotes terms that occur frequently in conjunction with each other – like "Hitler" and "Mussolini." Indirectly associated terms, on the other hand, denote such terms that imply a relationship when mediated through some other intermediate term. An example in this regard would be "Hitler," "Mussolini," and "Germany." This has resulted in GloVe embeddings emerging as one of the best embeddings, especially in regard to applications like evaluating word similarity and solving analogies.

Several empirical studies have shown that GloVe embeddings are useful for various downstream NLP applications. For example, Pennington et al. (2014) compared GloVe to Word2Vec on several benchmarks, including word similarity, word analogy, and named entity recognition tasks. On these benchmarks, the performance of GloVe often turned out to be as good as or even better than that of Word2Vec. To illustrate, on the WordSim-353 semantic similarity task, GloVe achieved higher correlation scores with human judgments than Word2Vec. Moreover, the global modeling methodology used by GloVe makes it particularly effective in tasks that require a comprehensive understanding of language, which includes topic modeling and document classification.

Despite the benefits, GloVe also has some specific limitations. Its greatest limitation would be on relying on a static co-occurrence matrix which might not be able to capture nuances in many contexts.

For example, the word "bank" would share only one embedding regardless of referring to a financial institution or bank of a river. The second problem is that just like Word2Vec, GloVe has problems managing OOV words since embedding is precomputed and then does not adapt dynamically. This problem is most significant when dealing with morphologically rich languages. Further limitations in the case of GloVe have to do with its co-occurrence matrix, or at least in its dependency on its accuracy. In less representative and less balanced corpora, the model is very prone to the lack of generalization when generating its embeddings.

The performance of GloVe relies on the linguistic features of the target language in multilingual and cross-lingual scenarios. A comparative study on GloVe against Word2Vec and FastText for English and French, showed that although GloVe excels at representing the global relationship

between words, it performs poorly in languages that have complex morphology since the subword information becomes critical there (Grave et al., 2018). Therefore, embedding techniques have to be chosen based on the linguistic requirements of specific NLP applications. In summary, GloVe is a very strong approach to generating word embeddings that rely on global co-occurrence statistics to produce rich semantic properties in the embeddings. While its static nature and reliance on precomputed co-occurrence matrices pose challenges, GloVe remains a useful tool for tasks requiring a comprehensive understanding of language.

FastText

FastText was introduced at Facebook AI Research by Bojanowski et al. in 2017. It improves upon the Word2Vec framework by incorporating sub-word level data into the generation of word embeddings. Unlike words, which are considered atomic units in traditional models including Word2Vec and GloVe, FastText words are considered as aggregations of character n-grams. This approach makes embedding creation more morphologically dependent, thus making it significantly better for handling OOVs and languages with complex morphologies.

At its core, FastText characterizes words through an aggregation of their underlying n-grams. For example, the word "playing" can be broken down into character n-grams such as <pla>, <lay>, <ayi>, and so on, using specific boundary indicators (< and >). The embedding associated with the word is then determined by summing the embeddings of these n-grams. Using this subword structure, FastText can construct a meaningful representation of rare or unknown words from the constituent parts. This characteristic addresses the major limitation that Word2Vec and GloVe both have: failure to manage OOV words properly.

FastText uses the same paradigm of prediction as Word2Vec; thus, it is feasible to use the CBOW and Skip-Gram models.

However, addition of subword information fundamentally improves its ability to capture linguistic features, especially in handling languages that contain rich morphologies or compounding features. Using CBOW, FastText predicts a target word based upon the embeddings of its adjacent words and their constituent ngrams. Meanwhile, using Skip-Gram, a target word, along with all of its subword components, are used to predict the surrounding words. An important advantage of FastText is its capacity to represent words that are similar in root or prefix with high effectiveness. For example, in languages like French or Finnish, where morphology is complex, terms like "jouer" and "jouant," which mean "to play" and "playing," respectively, contain many common subwords. FastText learns this common structure well so it can generate embeddings that can represent these relationships without having a huge corpus of fully observed word forms. The implementation of subword modeling significantly enhances

performance in multilingual settings where language variability is often encountered as a problem for standard word embedding models.

Empirical evaluations of FastText have demonstrated that it outperforms Word2Vec and GloVe in many NLP tasks, especially on low-resource languages or large OOV problems. Grave et al. (2018) did a deep analysis of FastText Embeddings across 157 languages and found that it was able to work much more robustly even in areas where training data were sparse. In particular, when deadlingwith part-of-speech tagging and named entity recognition, FastText was found to perform much better than Word2Vec and GloVe. It was able to detect morphological subtlety and minimize the impact of OOV words. FastText works well for cross-lingual applications. Joulin et al. (2018) augmented FastText to achieve multilingual embeddings by aligning monolingual spaces using a limited bilingual lexicon. The alignment enabled FastText to work on tasks, such as machine translation and cross-lingual information retrieval, with a minimal requirement of additional training data, therefore emphasizing its adaptability to different linguistic contexts.

Despite all these benefits, FastText also has some limitations.

A major drawback is its increased computational complexity in both training and inference compared to Word2Vec, since the model needs to take into consideration the embeddings of all constituting n-grams of every single word. This leads to increased memory requirements as well as longer processing times, especially in applications containing large vocabularies. Moreover, although FastText demonstrates proficiency in encapsulating subword information, it retains the inherent static characteristic of embeddings used in Word2Vec and GloVe, which involves assigning a singular vector to a word without regard for its contextual usage. Consequently, it does not possess the contextual sensitivity observed in contemporary transformer-based architectures like BERT or GPT. Furthermore, while it gives sturdy performance for morphologically rich languages, the reliance on n-grams may infuse noise into languages which make use of non-concatenative morphology or languages which extensively use homonyms.

In a nutshell, FastText represents one of the most important contributions to the area of word embedding technology by including information at the subword level into representations. Its possibility of producing embeddings for out-of-vocabulary words besides robustness over diverse languages make it a powerful tool for NLP tasks in either high-resource or low-resource settings. However, this static nature and computational cost reminds one to be prudent in considerations of use cases while making a choice for an embedding technique. FastText is and continues to be an indispensable part of the natural language toolkit, especially in environments with high morphological complexity and linguistic diversity.

Comparative Analysis

Support for multiple languages is one of the main areas of research with respect to different vector embeddings. This can be attributed to the increased demands for natural language processing systems that work efficiently across a range of languages. Each of these models can generate embeddings for multiple languages, but their effectiveness often varies based on linguistic features, the availability of training data, and specific task requirements. Word2Vec and GloVe, which have static embeddings, struggle when used in morphologically dense languages, in which the words can have different forms but share the same origin. On the other hand, FastText, because it can use subword information, offers an added advantage in these scenarios to better represent the morphologically complex words and also to represent OOV words.

Comparative studies have identified the strengths and weaknesses of these models in multilingual applications. For example, Grave et al. (2018) evaluated Word2Vec, GloVe, and FastText on cross-lingual part-of-speech tagging and named-entity recognition tasks in English, French, and Hindi, among other languages. Their results showed that FastText outperformed both Word2Vec and GloVe in all languages whose morphology is complex, as is the case with Hindi and Finnish, due to its subword modeling approach. However, in low-morphology languages with large amounts of training data, such as English, GloVe was on par and even performed better in document classification, a task requiring global semantic relationships. Word2Vec proved to be very consistent across tasks but fell behind FastText when it came to low-resource or morphologically complex languages, where the inability to handle OOV words presented a major problem.

In applications involving multiple languages, including machine translation and multilingual sentiment analysis, FastText has shown a significant advantage due to its intrinsic ability in harmonizing embeddings across different languages. Joulin et al. (2018) improved FastText to align monolingual embeddings by using a small bilingual lexicon, which allowed cross-lingual word similarity and translation activities with minimal additional data. However, Word2Vec and GloVe required larger datasets and more sophisticated alignment techniques to get comparable results and were therefore not seen as practical in low-resource settings.

Despite its benefits, FastText has some challenges in multilingual settings. Its reliance on n-grams can sometimes bring noise into languages that do not use concatenative morphology, such as Arabic and Hebrew, where the connections between meanings are not exclusively tied to linear subword structures. Additionally, in languages with rich resources, GloVe's co-occurrence-based embeddings often capture more general semantic properties better than FastText, making it a more appropriate choice for tasks like topic modeling and document clustering.

In conclusion, though all these three models have significantly enhanced multilingual NLP performance, their relative strengths justify using them for different sets of scenarios. FastText's subword modeling thus makes it much more fit to handle OOVs and morphologically rich languages and is inescapable for low-resource tasks and cross-linguality. In contrast, GloVe is more powerful in representing global semantic relationships in high-resource environments, while Word2Vec offers a judiciously balanced and computationally efficient alternative for tasks that have access to extensive corpora and uncomplicated linguistic features. The choice of the right model fundamentally depends on the specific linguistic and computational requirements of the particular application.

Key Variables & Research Objectives

Two key variables are going to be examined through the course of this study – the downstream NLP task which we are performing using the embeddings, and the language of the dataset. Our objectives in this study will be to:

1. Explore how different word embeddings perform across different languages which vary in syntactic/semantic structure and morphological complexity
2. Examine the performance of different word embeddings across three downstream NLP applications – sentiment analysis, part-of-speech tagging, and named-entity recognition
3. Determine which word embedding is best suited for a general-purpose, multi-language NLP system based on consistency across a variety of tasks.

The first variable is the NLP task which we are testing the different word embeddings on. The same implementation of each task will be used – that is to say, the library from which the model is acquired, the loss function used, and other characteristics will remain consistent for a given task. This will ensure consistency in the results. It is important that the word embedding works across a variety of given downstream tasks so as to be used for general-purpose NLP systems.

In terms of the specific tasks at hand, GloVe is predicted to outperform the other models in sentiment analysis

The second variable to keep in mind is the language selected. Different languages have different degrees of morphological complexity, alongside different syntactic and semantic relations. For the purposes of our study, I are taking into consideration an Indo-European language, French, and a Germanic language, English. While both languages share many similarities such as SVO order and many shared lexical items due to their close proximity, they still have distinct inflectional styles. Moreover, gender plays a much larger role in the French language than it does in English. As is seen in the literature review, there is a growing need for multi-language NLP systems and therefore, I sought to examine which vector embedding would be best suited for use across various languages.

GloVe and Word2Vec will perform similarly when it comes to POS tagging and sentiment analysis in English as compared to FastText. This is due to the tasks being heavily dependent on syntactic and semantic patterns, which are better identified by the former two embeddings. Between GloVe and Word2Vec, GloVe is expected to perform slightly better than Word2Vec because of its global co-occurence modeling, which captures broader semantic relationships. Meanwhile, FastText will perform better in terms of named-entity recognition in English due to the higher frequency of rare and out-of-vocabulary words and phrases. FastText's ability to take n-grams into consideration gives it an edge over GloVe and Word2Vec in this context.

In French, meanwhile, FastText is expected to outperform both the other models across all tasks. The rationale behind this is the comparatively more complex morphology of French compared to English. Once again, as FastText considers n-grams and subword components, it does a better job with morphologically rich languages.

Alongisde analyzing which embedding is best for a given task, we will also be analyzing which embedding is most consistent across the 3 tasks. This will allow us to determine which word embedding would be best for a general-purpose NLP application.

FastText is hypothesized to be the most consistent model across tasks and languages. Its subword approach is not only extremely beneficial for analyzing morphologically complex languages, but it also reduces dependence on large corpora, making it adaptable to different datasets. Conversely, the dependence of GloVe on global co-occurrence statistics makes it prone to variability with a change in the quality of training data or corpus size, while Word2Vec's inability to handle out-of-vocabulary words introduces inconsistencies in low-resource settings or languages with significant morphological variability.

Core Hypothesis Statement

In POS tagging and sentiment analysis with an English dataset, GloVe is expected to outperform the other two embeddings due to its global co-occurrence modelling. Meanwhile, FastText is expected to outperform GloVe and Word2Vec in English named-entity recogntiion because of its handling of out-of-vocabulary and rare words. FastText is also expected to be the best-performing word embeddings with French datasets due to its subword modeling, which makes it more suitable for more morphologically complex structure. Overall, FastText is also expected to be the most consistent word embedding, outperforming the alternatives due to its enhanced subword modeling and handling of rare and out-of-vocabulary words.

Methodology

Datasets

In order to ensure that all vector embeddings were compared with the same parameters for each task, the same training and testing data from the same datasets was used. This also ensured direct control over the data being used to train and test the models. The links to the datasets can be found in the appendix.

For all tasks, there was about an 80-20 split performed with respect to the training and testing data.

The dataset used for POS tagging in English is the English portion of the Parallel Universal Dependencies (PUD) treebanks created for the CoNLL 2017 shared task on Multilingual Parsing It consists of 1,117 sentences and 21,656 tokens. Meanwhile, for POS tagging in French, the dataset used is an automatic conversion of the French QuestionBank v1, a corpus entirely made of questions. It consists of 2,289 sentences and 24,452 tokens.

The "Large Movie Review Dataset" was used for sentiment analysis in English. The dataset is compiled from a collection of 50,000 reviews from IMDB on the condition there are no more than 30 reviews per movie. Negative reviews have scores less or equal than 4 out of 10 while a positive review have scores greater or equal than 7 out of 10 while neutral reviews are not included. For our purposes, I used 25,000 reviews for training and 6,250 reviews for testing. The numbers of positive and negative reviews are equal.

Meanwhile, the AlloCine movie reviews dataset was used for sentiment analysis in French. It contains 100,000 positive and 100,000 negative reviews divided into 3 balanced splits. Similar to English, I used 25,000 reviews for training and 6,250 reviews for testing. The numbers of positive and negative reviews are equal.

To test named-entity recognition in English, I used the CoNLL-2003 Dataset. It consists of 3,684 sentences and 46,666 tokens for the testing data, and it contains 14,987 sentences and 204,567 tokens for the training data.

Meanwhile, to test named-entity recognition in French, I used the Babelscape Dataset. The data was presented in a csv file, which was converted into textual data which was then split according using the 80-20 split methodology. It consists of 2,536 sentences and 61,352 tokens for the testing data, and it contains 10,142 sentences and 244,146 tokens for the training data.

Use of BiLSTM Layer

A Bidirectional Long Short-Term Memory (BiLSTM) layer was added on top of the word embeddings for all models for all tasks. The decision to use a BiLSTM layer was motivated by its high efficacy in downstream NLP tasks mainly due to its ability to accurately understand both past and future data.

In NLP tasks such as POS tagging, NER, and sentiment analysis, the meaning of a word is highly dependent on its surrounding words. The BiLSTM processes the input sequence in both forward and backward directions, thus ensuring that the model is aware of context from both sides of a given word. For example, in the sentence "The bank on the river is beautiful," the word "bank" can only be interpreted correctly with information from both the preceding ("The bank on") and succeeding ("on the river") words.

In general, BiLSTMs are considered better than traditional unidirectional models like recurrent neural networks (RNNs) with respect to keeping track of long-term dependencies. Unidirectional models usually fail to preserve information over long sequences due to the vanishing gradient problem. Since BiLSTMs process sequences in two directions, this issue is avoided, and local and global dependencies are properly captured. For example, in sentiment analysis, a word can express a certain sentiment, depending on the words far apart in a sentence.

Coupled with embeddings such as Word2Vec, GloVe, or FastText, the BiLSTM layers extend the representation capability of the model. The embeddings present dense representations of words while providing contextual sequential information through BiLSTMs, allowing the model to make effective use of semantic as well as syntactic information.

In terms of the specific tasks which are covered in our study, BiLSTM is also seen as the best choice. Specifically, for part-of-speech tagging, the bidirectional processing helps in accurately determining the grammatical role of words by understanding contextual cues from both directions. For named-entity recognition, BiLSTMs allow the model to carefully consider the position and role of an entity within each sentence, improving the model’s ability to classify entities accurately. Lastly, for sentiment analysis, the BiLSTM captures nuanced sentiment shifts that may depend on the full sentence structure.

Libraries

For sentiment analysis and named-entity recognition, I used Tensorflow – specifically, tensorflow.keras. Meanwhile, for POS tagging, I used PyTorch. As the models being run were the same and they remained consistent across tasks and languages, I used the opportunity to explore and learn different libraries without compromising the integrity of the project.

I had previous experience using PyTorch for POS tagging from my CL1 project, which I built upon by using vector embeddings while keeping the core library usage consistent.

Results

POS Tagging in English

Word2Vec

Classification Report:

	precision	recall	f1-score	support
ADJ	0.67	0.68	0.67	313
ADP	0.89	0.90	0.90	499
ADV	0.74	0.76	0.75	192
AUX	0.86	1.00	0.93	217
CCONJ	1.00	1.00	1.00	111
DET	0.99	0.97	0.98	412
INTJ	0.00	0.00	0.00	0
NOUN	0.76	0.79	0.77	797
NUM	0.66	0.77	0.71	79
PART	0.64	0.83	0.72	103
PRON	0.97	0.91	0.94	172
PROPN	0.65	0.57	0.61	338
PUNCT	1.00	1.00	1.00	474
SCONJ	0.56	0.36	0.44	55
SYM	1.00	0.88	0.93	8
VERB	0.76	0.69	0.72	457
X	0.33	0.17	0.22	6
accuracy			0.82	4233
macro avg	0.73	0.72	0.72	4233
weighted avg	0.82	0.82	0.82	4233

Confusion Matrix:

	ADJ	ADP	ADV	AUX	CCONJ	DET	INTJ	...	PRON	PROPN	PUNCT	SCONJ	SYM	VERB	X
ADJ	213	3	8	0	0	0	0	...	0	22	0	1	0	12	0
ADP	1	450	0	0	0	0	0	...	0	1	0	0	0	0	0
ADV	15	8	145	0	0	1	0	...	1	9	0	0	0	5	0
AUX	0	0	0	217	0	0	0	...	0	0	0	0	0	0	0
CCONJ	0	0	0	0	111	0	0	...	0	0	0	0	0	0	0
DET	2	0	4	0	0	399	0	...	3	0	0	3	0	0	0
INTJ	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
NOUN	40	3	6	0	0	1	0	...	0	49	0	1	0	59	0
NUM	2	0	1	0	0	0	0	...	1	5	0	0	0	1	0
PART	0	1	17	0	0	0	0	...	0	0	0	0	0	0	0
PRON	0	0	0	0	0	4	0	...	157	0	0	10	0	0	0
PROPN	23	1	11	0	0	0	0	...	0	192	0	1	0	22	2
PUNCT	0	0	0	0	0	0	0	...	0	0	474	0	0	0	0
SCONJ	0	33	1	0	0	0	0	...	0	0	0	20	0	0	0
SYM	0	0	0	0	0	0	0	...	0	1	0	0	7	0	0
VERB	22	4	2	35	0	0	0	...	0	16	0	0	0	316	0
X	1	0	0	0	0	0	0	...	0	1	0	0	0	1	1

GloVe

Classification Report:

	precision	recall	f1-score	support
ADJ	0.83	0.80	0.82	313
ADP	0.91	0.90	0.90	499
ADV	0.84	0.82	0.83	192
AUX	0.87	0.97	0.92	217
CCONJ	0.99	1.00	1.00	111
DET	0.98	0.98	0.98	412
INTJ	0.00	0.00	0.00	0
NOUN	0.87	0.91	0.89	797
NUM	0.89	0.99	0.93	79
PART	0.65	0.99	0.79	103
PRON	0.90	0.97	0.93	172
PROPN	0.94	0.94	0.94	338
PUNCT	1.00	1.00	1.00	474
SCONJ	1.00	0.16	0.28	55
SYM	1.00	0.88	0.93	8
VERB	0.89	0.77	0.83	457
X	0.00	0.00	0.00	6
accuracy			0.90	4233
macro avg	0.80	0.77	0.76	4233
weighted avg	0.90	0.90	0.90	4233

Confusion Matrix:

	ADJ	ADP	ADV	AUX	CCONJ	DET	INTJ	NOUN	NUM	PART	PRON	PROPN	PUNCT	SCONJ	SYM	VERB	X
ADJ	251	3	23	0	0	0	0	24	0	0	0	6	0	0	0	6	0
ADP	0	450	0	0	0	0	0	2	0	46	0	0	0	0	0	1	0
ADV	9	8	158	0	0	4	0	5	0	7	1	0	0	0	0	0	0
AUX	0	0	0	210	0	0	0	0	0	0	0	0	0	0	0	7	0
CCONJ	0	0	0	0	111	0	0	0	0	0	0	0	0	0	0	0	0
DET	2	0	0	0	1	402	0	1	0	0	6	0	0	0	0	0	0
INTJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
NOUN	22	0	5	0	0	0	0	723	8	0	0	11	0	0	0	28	0
NUM	0	0	0	0	0	0	0	0	78	0	1	0	0	0	0	0	0
PART	0	1	0	0	0	0	0	0	0	102	0	0	0	0	0	0	0
PRON	0	0	0	0	0	4	0	0	1	0	167	0	0	0	0	0	0
PROPN	8	0	0	0	0	0	0	11	0	0	0	318	0	0	0	1	0
PUNCT	0	0	0	0	0	0	0	0	0	0	0	0	474	0	0	0	0
SCONJ	0	33	2	0	0	0	0	0	0	0	11	0	0	9	0	0	0
SYM	0	0	0	0	0	0	0	0	1	0	0	0	0	0	7	0	0
VERB	10	2	0	31	0	0	0	60	0	0	0	1	0	0	0	353	0
X	0	0	0	0	0	0	0	1	0	1	0	4	0	0	0	0	0

FastText

Classification Report:

	precision	recall	f1-score	support
ADJ	0.81	0.64	0.72	313
ADP	0.91	0.89	0.90	499
ADV	0.92	0.70	0.80	192
AUX	0.86	1.00	0.93	217
CCONJ	0.99	0.99	0.99	111
DET	0.99	0.97	0.98	412
INTJ	0.00	0.00	0.00	0
NOUN	0.63	0.92	0.75	797
NUM	0.92	0.77	0.84	79
PART	0.65	0.99	0.79	103
PRON	0.97	0.91	0.94	172
PROPN	0.99	0.45	0.62	338
PUNCT	1.00	1.00	1.00	474
SCONJ	0.55	0.40	0.46	55
SYM	1.00	0.88	0.93	8
VERB	0.86	0.68	0.76	457
X	0.00	0.00	0.00	6
accuracy			0.83	4233
macro avg	0.77	0.72	0.73	4233
weighted avg	0.86	0.83	0.83	4233

Confusion Matrix:

	ADJ	ADP	ADV	AUX	CCONJ	DET	INTJ	NOUN	NUM	PART	PRON	PROPN	PUNCT	SCONJ	SYM	VERB	X
ADJ	201	3	6	0	0	0	0	102	0	0	0	0	0	0	0	1	0
ADP	0	446	0	0	0	0	0	3	0	46	0	0	0	4	0	0	0
ADV	17	7	135	0	0	1	0	23	0	7	1	0	0	1	0	0	0
AUX	0	0	0	217	0	0	0	0	0	0	0	0	0	0	0	0	0
CCONJ	0	0	0	0	110	1	0	0	0	0	0	0	0	0	0	0	0
DET	2	0	4	0	1	398	0	1	0	0	3	0	0	3	0	0	0
INTJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
NOUN	10	0	0	0	0	0	0	735	4	0	0	1	0	0	0	47	0
NUM	0	0	0	0	0	0	0	17	61	0	1	0	0	0	0	0	0
PART	0	1	0	0	0	0	0	0	0	102	0	0	0	0	0	0	0
PRON	0	0	0	0	0	4	0	0	1	0	157	0	0	10	0	0	0
PROPN	7	0	0	0	0	0	0	178	0	0	0	153	0	0	0	0	0
PUNCT	0	0	0	0	0	0	0	0	0	0	0	0	474	0	0	0	0
SCONJ	0	31	1	0	0	0	0	1	0	0	0	0	0	22	0	0	0
SYM	0	0	0	0	0	0	0	1	0	0	0	0	0	0	7	0	0
VERB	10	2	0	35	0	0	0	97	0	0	0	0	0	0	0	313	0
X	0	0	0	0	0	0	0	3	0	1	0	1	0	0	0	1	0

POS Tagging in French

Word2Vec

Classification Report:

	precision	recall	f1-score	support
ADJ	0.72	0.47	0.57	295
ADP	0.99	0.99	0.99	551
ADV	0.98	0.95	0.97	174
AUX	0.89	1.00	0.94	315
CCONJ	1.00	1.00	1.00	14
DET	0.88	0.99	0.93	800
INTJ	0.00	0.00	0.00	0
NOUN	0.82	0.84	0.83	812
NUM	0.79	0.50	0.61	22
PRON	0.90	0.99	0.94	343
PROPN	0.73	0.73	0.73	439
PUNCT	1.00	1.00	1.00	474
SCONJ	1.00	0.08	0.15	37
SYM	0.00	0.00	0.00	1
VERB	0.88	0.78	0.83	386
X	0.51	0.44	0.47	48
_	0.88	1.00	0.94	180
accuracy			0.88	4891
macro avg	0.76	0.69	0.70	4891
weighted avg	0.87	0.88	0.87	4891

Confusion Matrix:

	ADJ	ADP	ADV	AUX	CCONJ	DET	INTJ	NOUN	NUM	PRON	PROPN	PUNCT	SCONJ	SYM	VERB	X	_
ADJ	139	0	0	0	0	100	0	29	0	0	20	0	0	0	4	3	0
ADP	1	544	0	0	0	0	0	1	1	0	2	0	0	0	0	1	1
ADV	0	0	166	0	0	0	0	5	0	0	1	0	0	0	0	0	2
AUX	0	0	0	315	0	0	0	0	0	0	0	0	0	0	0	0	0
CCONJ	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0
DET	0	0	0	0	0	789	0	0	0	3	0	0	0	0	0	0	8
INTJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
NOUN	26	2	1	1	0	1	0	680	2	0	61	0	0	0	25	4	9
NUM	2	0	0	0	0	0	0	4	11	0	5	0	0	0	0	0	0
PRON	1	0	0	0	0	2	0	0	0	338	1	0	0	0	0	0	1
PROPN	19	2	3	0	0	0	0	72	0	0	321	0	0	0	11	11	0
PUNCT	0	0	0	0	0	0	0	0	0	0	0	474	0	0	0	0	0
SCONJ	0	0	0	0	0	0	0	0	0	33	0	0	3	0	0	0	1
SYM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
VERB	0	0	0	38	0	0	0	27	0	0	18	0	0	0	300	1	2
X	5	0	0	0	0	0	0	10	0	0	11	0	0	0	1	21	0
_	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	180

GloVe

Classification Report:

	precision	recall	f1-score	support
ADJ	0.79	0.48	0.60	295
ADP	1.00	0.99	0.99	551
ADV	0.98	0.95	0.97	174
AUX	0.89	1.00	0.94	315
CCONJ	1.00	1.00	1.00	14
DET	0.88	0.99	0.93	800
INTJ	0.00	0.00	0.00	0
NOUN	0.85	0.86	0.86	812
NUM	0.84	0.95	0.89	22
PRON	0.90	0.99	0.94	343
PROPN	0.78	0.77	0.77	439
PUNCT	1.00	1.00	1.00	474
SCONJ	1.00	0.08	0.15	37
SYM	0.00	0.00	0.00	1
VERB	0.91	0.78	0.84	386
X	0.34	0.48	0.40	48
_	0.88	1.00	0.94	180
accuracy			0.89	4891
macro avg	0.77	0.72	0.72	4891
weighted avg	0.89	0.89	0.88	4891

Confusion Matrix:

	ADJ	ADP	ADV	AUX	CCONJ	DET	INTJ	NOUN	NUM	PRON	PROPN	PUNCT	SCONJ	SYM	VERB	X	_
ADJ	143	0	1	0	0	100	0	34	1	0	14	0	0	0	0	2	0
ADP	1	544	1	0	0	0	0	1	0	0	2	0	0	0	1	0	1
ADV	0	0	166	0	0	0	0	2	0	0	2	0	0	0	0	2	2
AUX	0	0	0	315	0	0	0	0	0	0	0	0	0	0	0	0	0
CCONJ	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0
DET	0	0	0	0	0	789	0	0	0	3	0	0	0	0	0	0	8
INTJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
NOUN	19	0	1	1	0	1	0	700	3	1	48	0	0	0	13	16	9
NUM	0	0	0	0	0	0	0	0	21	0	1	0	0	0	0	0	0
PRON	1	0	0	0	0	2	0	0	0	338	1	0	0	0	0	0	1
PROPN	12	0	0	0	0	0	0	57	0	0	337	0	0	0	13	20	0
PUNCT	0	0	0	0	0	0	0	0	0	0	0	474	0	0	0	0	0
SCONJ	0	0	0	0	0	0	0	0	0	33	0	0	3	0	0	0	1
SYM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
VERB	5	0	0	38	0	0	0	18	0	0	17	0	0	0	301	5	2
X	1	0	0	0	0	0	0	13	0	0	10	0	0	0	1	23	0
_	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	180

FastText

Classification Report:

	precision	recall	f1-score	support
ADJ	0.95	0.45	0.61	295
ADP	1.00	0.99	0.99	551
ADV	1.00	0.95	0.98	174
AUX	0.89	1.00	0.94	315
CCONJ	1.00	1.00	1.00	14
DET	0.89	0.99	0.93	800
INTJ	0.00	0.00	0.00	0
NOUN	0.97	0.74	0.84	812
NUM	1.00	0.50	0.67	22
PRON	0.90	0.99	0.94	343
PROPN	0.56	0.99	0.72	439
PUNCT	1.00	1.00	1.00	474
SCONJ	1.00	0.08	0.15	37
SYM	0.00	0.00	0.00	1
VERB	1.00	0.77	0.87	386
X	0.90	0.40	0.55	48
_	0.88	1.00	0.94	180
accuracy			0.88	4891
macro avg	0.82	0.70	0.71	4891
weighted avg	0.91	0.88	0.88	4891

Confusion Matrix:

	ADJ	ADP	ADV	AUX	CCONJ	DET	INTJ	NOUN	NUM	PRON	PROPN	PUNCT	SCONJ	SYM	VERB	X	_
ADJ	134	0	0	0	0	100	0	11	0	0	50	0	0	0	0	0	0
ADP	0	544	0	0	0	0	0	0	0	0	6	0	0	0	0	0	1
ADV	0	0	166	0	0	0	0	0	0	0	6	0	0	0	0	0	2
AUX	0	0	0	315	0	0	0	0	0	0	0	0	0	0	0	0	0
CCONJ	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0
DET	0	0	0	0	0	789	0	0	0	3	0	0	0	0	0	0	8
INTJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
NOUN	6	0	0	1	0	0	0	603	0	0	192	0	0	0	1	0	9
NUM	0	0	0	0	0	0	0	0	11	0	11	0	0	0	0	0	0
PRON	1	0	0	0	0	2	0	0	0	338	1	0	0	0	0	0	1
PROPN	0	0	0	0	0	0	0	4	0	0	434	0	0	0	0	1	0
PUNCT	0	0	0	0	0	0	0	0	0	0	0	474	0	0	0	0	0
SCONJ	0	0	0	0	0	0	0	0	0	33	0	0	3	0	0	0	1
SYM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
VERB	0	0	0	38	0	0	0	4	0	0	45	0	0	0	296	1	2
X	0	0	0	0	0	0	0	0	0	0	29	0	0	0	0	19	0
_	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	180

Sentiment Analysis in English

NOTE: (0,0) in the confusion matrix represents negative reviews which were actually found to be negative and (1,1) in the confusion matrix represents positive reviews which were actually found to be positive. Meanwhile, (0,1) represents positive reviews which were found to be negative and (1,0) represents negative reviews which were found to be positive

Word2Vec

```
Confusion Matrix
[[2649  476]
 [ 825 2300]]

Classification Report
              precision    recall  f1-score   support

    Negative       0.76      0.85      0.80      3125
    Positive       0.83      0.74      0.78      3125

    accuracy                           0.79      6250
   macro avg       0.80      0.79      0.79      6250
weighted avg       0.80      0.79      0.79      6250
```

GloVe

```
Confusion Matrix
[[2279  846]
 [ 427 2698]]

Classification Report
              precision    recall  f1-score   support

    Negative       0.84      0.73      0.78      3125
    Positive       0.76      0.86      0.81      3125

    accuracy                           0.80      6250
   macro avg       0.80      0.80      0.80      6250
weighted avg       0.80      0.80      0.80      6250
```

FastText

```
Confusion Matrix
[[2609  516]
 [1126 1999]]

Classification Report
              precision    recall  f1-score   support

    Negative       0.70      0.83      0.76      3125
    Positive       0.79      0.64      0.71      3125

    accuracy                           0.74      6250
   macro avg       0.75      0.74      0.73      6250
weighted avg       0.75      0.74      0.73      6250
```

Sentiment Analysis in French

NOTE: (0,0) in the confusion matrix represents negative reviews which were actually found to be negative and (1,1) in the confusion matrix represents positive reviews which were actually found to be positive. Meanwhile, (0,1) represents positive reviews which were found to be negative and (1,0) represents negative reviews which were found to be positive

Word2Vec

```
Confusion Matrix
[[2440  685]
 [ 460 2665]]

Classification Report
              precision    recall  f1-score   support

    Negative       0.84      0.78      0.81      3125
    Positive       0.80      0.85      0.82      3125

    accuracy                           0.82      6250
   macro avg       0.82      0.82      0.82      6250
weighted avg       0.82      0.82      0.82      6250
```

GloVe

```
Confusion Matrix
[[2078 1047]
 [ 419 2706]]

Classification Report
              precision    recall  f1-score   support

    Negative       0.83      0.66      0.74      3125
    Positive       0.72      0.87      0.79      3125

    accuracy                           0.77      6250
   macro avg       0.78      0.77      0.76      6250
weighted avg       0.78      0.77      0.76      6250
```

FastText

```
Confusion Matrix
[[2503  622]
 [ 807 2318]]

Classification Report
              precision    recall  f1-score   support

    Negative       0.76      0.80      0.78      3125
    Positive       0.79      0.74      0.76      3125

    accuracy                           0.77      6250
   macro avg       0.77      0.77      0.77      6250
weighted avg       0.77      0.77      0.77      6250
```

NER in English

Word2Vec

Confusion Matrix:

	O	B-PER	I-PER	B-ORG	I-ORG	B-LOC	I-LOC	B-MISC	I-MISC
O	250359	167	96	107	51	84	9	13	0
B-PER	698	817	29	54	5	7	0	0	0
I-PER	465	18	654	6	7	0	0	0	0
B-ORG	849	74	16	683	9	23	0	6	0
I-ORG	496	13	49	55	211	6	4	1	0
B-LOC	879	56	2	60	3	656	0	4	0
I-LOC	187	5	6	2	13	7	34	1	1
B-MISC	631	14	2	8	4	6	1	36	0
I-MISC	207	0	1	1	5	1	1	0	0

Classification Report

	precision	recall	f1-score	support
O	0.98	1.00	0.99	250886
B-PER	0.70	0.51	0.59	1610
I-PER	0.76	0.57	0.65	1150
B-ORG	0.70	0.41	0.52	1660
I-ORG	0.69	0.25	0.37	835
B-LOC	0.83	0.40	0.54	1660
I-LOC	0.69	0.13	0.22	256
B-MISC	0.59	0.05	0.09	702
I-MISC	0.00	0.00	0.00	216
accuracy			0.98	258975
macro avg	0.66	0.37	0.44	258975
weighted avg	0.97	0.98	0.97	258975

GloVe

Confusion Matrix:

	O	B-PER	I-PER	B-ORG	I-ORG	B-LOC	I-LOC	B-MISC	I-MISC
O	249367	246	89	398	135	335	26	230	60
B-PER	1095	198	58	192	13	45	2	7	0
I-PER	904	37	81	62	46	13	1	6	0
B-ORG	836	31	12	643	56	45	1	28	8
I-ORG	494	12	13	95	190	18	4	5	4
B-LOC	1231	33	11	110	14	225	12	20	4
I-LOC	173	5	10	8	13	8	32	3	4
B-MISC	456	14	7	16	2	23	0	168	16
I-MISC	149	2	0	3	1	4	1	5	51

Classification Report:

	precision	recall	f1-score	support
O	0.31	0.14	0.19	1660
B-PER	0.36	0.24	0.29	702
I-PER	0.42	0.39	0.40	1660
B-ORG	0.34	0.12	0.18	1610
I-ORG	0.41	0.12	0.19	256
B-LOC	0.35	0.24	0.28	216
I-LOC	0.40	0.23	0.29	835
B-MISC	0.29	0.07	0.11	1150
I-MISC	0.98	0.99	0.99	250886
accuracy			0.97	258975
macro avg	0.43	0.28	0.32	258975
weighted avg	0.96	0.97	0.96	258975

FastText

Confusion Matrix:

	O	B-PER	I-PER	B-ORG	I-ORG	B-LOC	I-LOC	B-MISC	I-MISC
O	250356	123	75	173	60	55	0	44	0
B-PER	886	525	20	129	9	36	0	5	0
I-PER	646	17	433	34	15	5	0	0	0
B-ORG	814	80	20	687	1	42	0	16	0
I-ORG	535	17	61	110	97	11	0	4	0
B-LOC	945	39	4	67	0	577	0	28	0
I-LOC	214	2	21	1	8	9	0	1	0
B-MISC	554	27	4	32	0	16	0	69	0
I-MISC	200	3	4	6	0	0	0	3	0

Classification Report

	precision	recall	f1-score	support
O	0.98	1.00	0.99	250886
B-PER	0.63	0.33	0.43	1610
I-PER	0.67	0.38	0.48	1150
B-ORG	0.55	0.41	0.47	1660
I-ORG	0.51	0.12	0.19	835
B-LOC	0.77	0.35	0.48	1660
I-LOC	0.00	0.00	0.00	256
B-MISC	0.41	0.10	0.16	702
I-MISC	0.00	0.00	0.00	216
accuracy			0.98	258975
macro avg	0.50	0.30	0.36	258975
weighted avg	0.97	0.98	0.97	258975

NER in French

Word2Vec

Confusion Matrix:

	B-LOC	B-MISC	B-ORG	B-PER	I-LOC	I-MISC	I-ORG	I-PER	O
B-LOC	467	5	0	43	0	2	0	9	1496
B-MISC	5	99	3	27	0	1	1	5	357
B-ORG	10	5	68	40	0	1	0	5	529
B-PER	1	3	0	758	0	2	1	6	508
I-LOC	45	0	0	10	5	7	0	45	396
I-MISC	4	16	1	18	0	249	7	41	458
I-ORG	14	0	0	25	3	14	124	51	776
I-PER	1	0	0	62	0	4	0	755	466
O	86	58	2	241	3	92	2	196	181466

Classification Report

	precision	recall	f1-score	support
B-LOC	0.74	0.23	0.35	2022
B-MISC	0.53	0.20	0.29	498
B-ORG	0.92	0.10	0.19	658
B-PER	0.62	0.59	0.61	1279
I-LOC	0.45	0.01	0.02	508
I-MISC	0.67	0.31	0.43	794
I-ORG	0.92	0.12	0.22	1007
I-PER	0.68	0.59	0.63	1288
O	0.97	1.00	0.98	182146
accuracy			0.97	190200
macro avg	0.72	0.35	0.41	190200
weighted avg	0.96	0.97	0.96	190200

GloVe

Confusion Matrix:

	B-LOC	B-MISC	B-ORG	B-PER	I-LOC	I-MISC	I-ORG	I-PER	O
B-LOC	1202	4	8	40	10	2	3	10	743
B-MISC	18	214	8	16	3	3	1	3	232
B-ORG	49	7	257	44	2	1	1	9	288
B-PER	22	2	1	809	4	6	2	21	412
I-LOC	52	0	0	6	155	9	5	46	235
I-MISC	10	3	0	16	5	409	4	33	314
I-ORG	18	0	9	22	14	24	518	76	326
I-PER	3	0	0	48	7	7	2	797	424
O	158	39	17	127	17	83	17	148	181540

Classification Report

	precision	recall	f1-score	support
B-LOC	0.78	0.59	0.68	2022
B-MISC	0.80	0.43	0.56	498
B-ORG	0.86	0.39	0.54	658
B-PER	0.72	0.63	0.67	1279
I-LOC	0.71	0.31	0.43	508
I-MISC	0.75	0.52	0.61	794
I-ORG	0.94	0.51	0.66	1007
I-PER	0.70	0.62	0.66	1288
O	0.98	1.00	0.99	182146
accuracy			0.98	190200
macro avg	0.80	0.56	0.64	190200
weighted avg	0.98	0.98	0.97	190200

FastText

Confusion Matrix:

	B-LOC	B-MISC	B-ORG	B-PER	I-LOC	I-MISC	I-ORG	I-PER	O
B-LOC	711	6	21	115	0	3	4	17	1145
B-MISC	15	104	8	44	0	4	0	9	314
B-ORG	25	9	108	75	0	0	1	8	432
B-PER	16	7	1	681	0	2	1	13	558
I-LOC	51	0	1	6	3	6	4	103	334
I-MISC	8	14	1	14	0	247	5	49	456
I-ORG	55	0	7	44	1	16	137	139	608
I-PER	2	0	0	40	0	13	0	687	546
O	127	43	27	255	0	77	9	260	181348

Classification Report

	precision	recall	f1-score	support
B-LOC	0.70	0.35	0.47	2022
B-MISC	0.57	0.21	0.31	498
B-ORG	0.62	0.16	0.26	658
B-PER	0.53	0.53	0.53	1279
I-LOC	0.75	0.01	0.01	508
I-MISC	0.67	0.31	0.43	794
I-ORG	0.85	0.14	0.23	1007
I-PER	0.53	0.53	0.53	1288
O	0.98	1.00	0.99	182146
accuracy			0.97	190200
macro avg	0.69	0.36	0.42	190200
weighted avg	0.96	0.97	0.96	190200

Discussion

Analysis with respect to key variables

POS Tagging in English

Word Embedding	Most Accurately Categorized Tags	Least Accurately Categorized Tags
Word2Vec	CCONJ (1.00), PUNCT (1.00), DET (0.99)	X (0.33), SCONJ (0.56), PROPN (0.65)
GloVe	PUNCT (1.00), SCONJ (1.00), SYM (1.00)	X (0.00), PART (0.65), ADJ (0.83)
FastText	PUNCT (1.00), SYM (1.00), PROPN (0.99)	SCONJ (0.55), PART (0.65), NOUN (0.63)

Table 1: Performance of Word Embeddings in POS Tagging with an English dataset with respect to precision

Word Embedding	Most Accurately Categorized Tags	Least Accurately Categorized Tags
Word2Vec	CCONJ (1.00), PUNCT (1.00), AUX (1.00)	X (0.17), SCONJ (0.36), PROPN (0.57)
GloVe	CCONJ (1.00), PUNCT (1.00), NUM (0.99)	X (0.00), SCONJ (0.16), VERB (0.77)
FastText	PUNCT (1.00), AUX (1.00), DET (0.97)	SCONJ (0.40), PROPN (0.45), ADJ (0.64)

Table 2: Performance of Word Embeddings in POS Tagging with an English dataset with respect to recall

Word Embedding	Most Accurately Categorized Tags	Least Accurately Categorized Tags
Word2Vec	CCONJ (1.00), PUNCT (1.00), DET (0.98)	X (0.22), SCONJ (0.44), PROPN (0.61)
GloVe	CCONJ (1.00), PUNCT (1.00), DET (0.98)	X (0.00), SCONJ (0.28), PART (0.79),
FastText	PUNCT (1.00), DET (0.98), AUX (0.93)	SCONJ (0.46), PROPN (0.62), PART (0.79)

Table 3: Performance of Word Embeddings in POS Tagging with an English dataset with respect to F1-score

The best model in terms of general accuracy was GloVe, which reported an accuracy of 0.90. This shows that the model has a strong understanding of general grammatical patterns with high precision, recall, and F1-scores for most common POS tags, such as adjectives, adverbs, nouns, and conjunctions. Even with the confusion matrix for GloVe, there were fewer confusions of similar tags in comparison to other models; this shows that GloVe possesses more subtle knowledge about the relationship of parts of speech.

While Word2Vec does well on frequent tags such as punctuation, conjunctions, and symbols, it is worse with less frequent POS categories. The likely reason is data sparsity; this is because the model may not have been trained enough to capture the finer details of more infrequent grammatical structures. Nonetheless, Word2Vec is still at an overall accuracy of 0.82 and thus can be useful in applications where the focus is on common POS tags.

In comparison, overall accuracy-wise, the winner is FastText, standing at 0.83. It really is particularly powerful when it comes to precision and recall for major parts-of-speech categories, like adjectives, adverbs, nouns, and particles. The strength of FastText is that, considering how it performs, it demonstrates a significant grasp of elementary grammatical features but at the same time possesses greater capacity when it comes to processing more infrequent parts-of-speech than Word2Vec.

This leaves the question of which is more suitable between these two models for the task of POS tagging based on specific needs for that particular application. In an effort to maximize common grammatical structure, it seems like GloVe could be best suited, but in order to not over-optimize to certain frequent but not so often used tags, FastText is better.

Word Embedding	Most Accurately Categorized Tags	Least Accurately Categorized Tags
Word2Vec	CCONJ (1.00), PUNCT (1.00), ADP (0.99)	X (0.51), NUM (0.79), PROPN (0.73)
GloVe	CCONJ (1.00), PUNCT (1.00), ADP (1.00)	X (0.34), PROPN (0.78), ADJ (0.79)
FastText	CCONJ (1.00), PUNCT (1.00), NUM (1.00)	PROPN (0.56), X (0.90), DET (0.89)

Table 4: Performance of Word Embeddings in POS Tagging with a French dataset with respect to precision

Word Embedding	Most Accurately Categorized Tags	Least Accurately Categorized Tags
Word2Vec	CCONJ (1.00), PUNCT (1.00), ADP (0.99)	SCONJ (0.08), X (0.44), NUM (0.50)
GloVe	CCONJ (1.00), PUNCT (1.00), AUX (1.00)	SCONJ (0.08), X (0.48), ADJ (0.48)
FastText	CCONJ (1.00), PUNCT (1.00), ADP (0.99)	SCONJ (0.08), X (0.40), NUM (0.50)

Table 5: Performance of Word Embeddings in POS Tagging with a French dataset with respect to recall

Word Embedding	Most Accurately Categorized Tags	Least Accurately Categorized Tags
Word2Vec	CCONJ (1.00), PUNCT (1.00), ADP (0.99)	SCONJ (0.15), X (0.47), NUM (0.61)
GloVe	CCONJ (1.00), PUNCT (1.00), ADP (0.99)	SCONJ (0.15), X (0.40), ADJ (0.60)
FastText	CCONJ (1.00), PUNCT (1.00), ADP (0.99)	SCONJ (0.15), X (0.55), NUM (0.67)

Table 6: Performance of Word Embeddings in POS Tagging with a French dataset with respect to F1-score

In general accuracy, the best model is GloVe with an accuracy of 0.89. This is a good indication that GloVe has a good hold on the general grammatical structure because it performs well in precision, recall, and F1-score for most of the commonly used POS tags like adjectives, adverbs, nouns, and conjunctions. The confusion matrix of GloVe also holds fewer misclassifications in similar tags than other models, meaning that it has more fine-grained knowledge regarding part-of-speech relations.

Word2Vec does a great job on the common tags such as punctuation, conjunctions, and symbols, but less well on the less common POS categories. This could be due to sparsity; the model might not have received enough training data to learn all the complexities of the less frequent grammatical structures. Nevertheless, Word2Vec still manages to maintain an overall accuracy of 0.88, making it a feasible choice for applications where common POS tags are the priority.

The FastText model strikes the middle ground of the two and yields the highest general accuracy at 0.88. It yields the best precision and recall for POS tags as important as adjectives, adverbs, nouns, and particles. FastText performance appears to be sound on essential grammatical structures and seems stronger in terms of less frequent part-of-speech categories than that of Word2Vec.

Finally, it will be at the discretion of the requirements that need to be satisfied on a particular task that between which of these word embeddings would be chosen for the purposes of POS tagging. GloVe would be ideal, for instance, in application if one is to emphasize maximization of accuracy, but in cases where such applications would demand a balanced approach able to handle frequent as well as infrequent POS tags, then FastText might suffice.

Sentiment Analysis in English

Overall, the models show fairly consistent performance with precision, recall, and F1-scores commonly in the 0.70-0.85 range.

Most balanced is the performance of the individual model in FastText. It has the precision, recall, and F1-scores similar for both classes, positive and negative; its macro and weighted-average F1-score is 0.73, indicating solid overall accuracy.

The Word2Vec model has a higher precision but lower recall on the positive class than on the negative class. This leads to a lower F1-score of 0.71 for the positive class, as opposed to 0.76 for the negative class. The macro and weighted avg F1-scores are at 0.73, which is equal to FastText.

The GloVe model shows the best overall performance, with maximum precision, recall, and F1-scores for positive and negative classes. Its macro and weighted avg F1-scores both are 0.80, that is the best among the three models.

Interpretation of results. Differences in model performance likely arise from the intrinsic characteristics of the word embeddings that each approach generates. FastText includes subword information, which can help it to be better at handling out-of-vocabulary terms and edge cases. Word2Vec is very local context-focused, and GloVe uses global co-occurrence statistics - the last two approaches likely are great at capturing semantic relationships in more common vocabulary.

Higher precision but lower recall for the positive class of the Word2Vec model implies this model might be conservative regarding the positive class and possibly might miss some true positives. FastText is better balanced in its performance; GloVe yields stronger results altogether, suggesting these models can strike the perfect balance between precision and recall.

Overall, it seems that GloVe will have the strongest performance based on given metrics.

Sentiment Analysis in French

The FastText model shows quite balanced performance of the positive and negative classes. All the precision, recall, and F1-scores lie in the 0.74-0.80 range, which implies solid overall accuracy. Also, the macro and weighted average F1-scores of 0.77 testify to the consistent nature of the model.

The Word2Vec model displays some dissimilarity in the treatment of positive versus negative sentiments. It gives a better precision for the positive class while the recall is poorer as compared to the negative class. Hence, it depicts a low F1-score for the positive class that stands at 0.82 as opposed to 0.81 for the negative. The macro and weighted average F1-scores are 0.82.

The best performance metrics on all fronts are found on the GloVe model. Precision, recall, and F1-scores were the highest of the three models for both positive and negative classes. The macro and weighted average F1-scores of 0.77 indicate excellent classification accuracy.

These differences in performance between the models can be seen to be arising from underlying characteristics of the word embeddings of each approach. It is thus likely that the advantage GloVe enjoys over capturing semantic relations could be attributed to the former's leverage of global co-occurrence statistics. Word2Vec, having its focus on local context, may make it somewhat conservative in the prediction of positive sentiment and hence yields lower recall for that class.

Overall, the GloVe model seems to be the best performer for this French sentiment analysis task, at least according to the metrics provided. Further testing on other datasets would be needed to draw more definite conclusions about the most appropriate model for a given use case.

NER in English

Word Embedding	Most Accurately Categorized Tags	Least Accurately Categorized Tags
Word2Vec	B-LOC (0.83), I-PER (0.76), B-PER (0.70)	B-MISC (0.59), I-LOC (0.69), I-ORG (0.69)
GloVe	I-MISC (0.98), I-PER (0.42), I-ORG (0.41)	B-MISC (0.29), B-ORG (0.34), B-LOC (0.35)
FastText	B-LOC (0.77), I-PER (0.67), B-PER (0.63)	B-MISC (0.41), I-ORG (0.51), B-ORG (0.55)

Table 7: Performance of Word Embeddings in NER with an English dataset with respect to precision

Word Embedding	Most Accurately Categorized Tags	Least Accurately Categorized Tags
Word2Vec	I-PER (0.57), B-PER (0.51), B-LOC (0.40)	B-MISC (0.05), I-MISC (0.00), I-LOC (0.13)
GloVe	I-MISC (0.99), I-PER (0.39), B-PER (0.24)	B-ORG (0.12), I-ORG (0.12), B-MISC (0.09)
FastText	I-PER (0.38), B-ORG (0.41), B-PER (0.33)	I-MISC (0.00), I-LOC (0.00), I-ORG (0.12)

Table 8: Performance of Word Embeddings in NER with an English dataset with respect to recall

Word Embedding	Most Accurately Categorized Tags	Least Accurately Categorized Tags
Word2Vec	I-PER (0.65), B-PER (0.59), B-LOC (0.54)	B-MISC (0.09), I-MISC (0.00), I-LOC (0.22)
GloVe	I-MISC (0.99), I-PER (0.40), B-PER (0.29)	B-MISC (0.11), B-ORG (0.18), I-ORG (0.19)
FastText	I-PER (0.48), B-LOC (0.48), B-ORG (0.47)	I-MISC (0.00), I-LOC (0.00), I-ORG (0.19)

Table 9: Performance of Word Embeddings in NER with an English dataset with respect to F1-score

The Word2Vec model is the one with the strongest performance overall, showing the highest accuracy, macro average F1-score, and weighted average F1-score across the various entity types. Its confusion matrix indicates that it has the capability to distinguish between most of the different entity types generally, since most of the diagonal values are the highest in each row, showing correct predictions. The model struggles the most with distinguishing between the "B-ORG" and "I-ORG" (organization) entities and the "B-MISC" and "I-MISC" (miscellaneous) entities, but it does a pretty good job overall in showing a strong understanding of the linguistic structures represented by the NER labels.

In comparison, the GloVe model shows slightly lower overall performance, with a lower accuracy and F1-scores. Its confusion matrix reveals more off-diagonal values, indicating more confusion between the different entity types. The model appears to have the most difficulty with the "I-LOC" (location) and "I-MISC" entities, often misclassifying them as other types.
The FastText model has the weakest performance of the three, with the lowest accuracy and F1-scores. Its confusion matrix contains the highest number of off-diagonal elements, which implies a higher rate of misclassification between the various types of entities. It seems that the model performs the worst on the "I-LOC", "I-MISC", and "I-ORG" entities, often confusing them with other types.

These differences in performance originate from the difference in linguistic and semantic representation that these word embeddings capture. It is likely that the overall strong performance of Word2Vec has led to stronger and more discriminative features for its NER task while better allowing it to distinguish the linguistic structures associated with the type of entity. Whereas, the GloVe as well as FastText may not correctly capture these linguistic nuances, which causes more confusion and misclassification between the entity types.

In the case of the NER task, the word structures that the entity types represent are probably better encoded by the contextual information modelled by Word2Vec. Word2Vec could be more sensitive to nuanced semantic differences between the different types of entities and hence outperform FastText on the NER classification task.

Furthermore, it seems that the FastText model is more challenged by certain types of entities, like "I-LOC", "I-MISC", and "I-ORG", which are often misclassified as other types of entities. This implies that the subword-level information captured by FastText is not as strong in distinguishing these more complex and ambiguous linguistic structures as the contextual information learned by Word2Vec.

In conclusion, from the analysis of classification reports and confusion matrices, the Word2Vec model appears to be the best model suited for the NER task, as it shows better linguistic structure understanding of the labels assigned by the NER.

NER in French

Word Embedding	Most Accurately Categorized Tags	Least Accurately Categorized Tags
Word2Vec	B-ORG (0.92), I-ORG (0.92), I-MISC (0.67)	I-LOC (0.45), B-MISC (0.53), B-PER (0.62)
GloVe	I-ORG (0.94), B-ORG (0.86), B-MISC (0.80)	I-LOC (0.71), I-PER (0.70), B-PER (0.72)
FastText	I-ORG (0.85), I-LOC (0.75), B-LOC (0.70)	B-PER (0.53), I-PER (0.53), B-MISC (0.57)

Table 10: Performance of Word Embeddings in NER with a French dataset with respect to precision

Word Embedding	Most Accurately Categorized Tags	Least Accurately Categorized Tags
Word2Vec	B-PER (0.59), I-PER (0.59), I-MISC (0.31)	I-LOC (0.01), I-ORG (0.12), B-ORG (0.10)
GloVe	B-LOC (0.59), B-PER (0.63), I-PER (0.62)	I-LOC (0.31), B-MISC (0.43), B-ORG (0.39)
FastText	B-PER (0.53), I-PER (0.53), I-MISC (0.31)	I-LOC (0.01), I-ORG (0.14), B-ORG (0.16)

Table 11: Performance of Word Embeddings in NER with a French dataset with respect to recall

Word Embedding	Most Accurately Categorized Tags	Least Accurately Categorized Tags
Word2Vec	I-PER (0.63), B-PER (0.61), I-MISC (0.43)	I-LOC (0.02), I-ORG (0.22), B-ORG (0.19)
GloVe	B-LOC (0.68), B-PER (0.67), I-PER (0.66)	I-LOC (0.43), B-MISC (0.56), B-ORG (0.54)
FastText	B-PER (0.53), I-PER (0.53), B-LOC (0.47)	I-LOC (0.01), I-ORG (0.23), B-ORG (0.26)

Table 12: Performance of Word Embeddings in NER with a French dataset with respect to F1-score

Overall, the GloVe model performs the best in all aspects. It has the highest precision scores ranging from 0.70 to 0.98 for the different types of entities. The recall scores are also very strong, with most of the entity types ranging between 0.31 to 0.63. This means that GloVe achieves the best F1-scores, with most of them ranging between 0.43 to 0.68.

The Word2Vec model shows more mixed results. While it has high precision for some of the entity types like I-ORG (0.92) and B-LOC (0.74), the recall is much lower, especially for location-based entities like B-LOC (0.23) and I-LOC (0.01). This imbalance leads to weaker overall F1-scores than GloVe.

The FastText model is middle of the road. While it doesn't show nearly the consistent performance of GloVe, it does appear to outperform Word2Vec; precision and recall values were generally in the 0.50 to 0.80 range. It didn't match the same high-water marks as GloVe, particularly for the more challenging entity types.

The macro and weighted average metrics have been analyzed further. GloVe had the highest macro F1-score at 0.64 while Word2Vec and FastText came in at 0.41 and 0.42 respectively. For the weighted averages that take into consideration class imbalances, it shows GloVe at 0.97, Word2Vec at 0.96, and FastText at 0.96.

These results indicate that the global co-occurrence information utilized by GloVe gives it an edge over the more local context modeling of Word2Vec. FastText's subword information offers a middle ground, improving on the shortcomings of Word2Vec but not quite reaching the overall strength of GloVe.

Overall, the GloVe model is the strongest performer for this French NER task based on the metrics provided.

Sources of Error/Bias

One of the major contributors of errors is the unbalancing and underrepresentation of data-sets. For example, in the case of POS tagging for French, the French QuestionBank only contains questions. Such a dataset tends to give a sharply bounded corpus for linguistic analysis yet turns out to be lesser in generalization to frequent uses of language. Similarly, Babelscape dataset for NER in French has comprehensive data and may be domain-biased. This bias might then skew the ability of models to generalize patterns across different text corpora. Characteristics of the dataset can lead to overfitting whereby models work well on test data but miss to capture wider patterns of linguistics.

The integrity and preliminary processing of datasets can also lead to potential errors. Preparatory steps, such as tokenization and lemmatization, are essential to properly prepare text before embedding and training models. However, errors or inconsistencies in these processes may be propagated into the models and produce incorrect or biased results. For instance, differences in handling contractions, hyphenated words, or punctuation can significantly affect the embeddings generated for these tokens, especially in morphologically complex languages like French.

Another source of error involves the static nature of embeddings themselves. Word2Vec, GloVe, and FastText generate static word representations because each word is assigned a single vector irrespective of its context in which it is being represented. This is especially worse for polysemous words like "bank" since one can refer to bank as a financial institution and it could also be referred to as the side of a river; the embeddings cannot identify context-dependent meanings. Hence, the models constructed using these embeddings may end up misclassifying words in sentiment analysis or NER applications, where such fine-grained contextual understanding is important.

Selection of evaluation metrics and even some nuances associated with a particular task may also have introduced errors.

For example, in NER, incorrect misclassification of entities (location vs. organization) rather than missing the fact that some entity is present may result from different implications, whereas their measurement in overall accuracy can be the same. Moreover, subtleties of certain tasks can refer to the hierarchical nature of some POS tags or the overlapping aspect of sentiment polarity of several reviews, which once more deceives the interpretation of the performance of a model.

Finally, the distinctive attributes of specific languages and the variability in their morphology present fundamental challenges. Although FastText's approach to subword modeling alleviates certain problems associated with the processing of morphologically intricate languages, it remains susceptible to inaccuracies stemming from noise within n-grams or poorly aligned subword configurations. Moreover, the dependence of GloVe and Word2Vec on global co-occurrence and local context, correspondingly, renders them less effective in languages such as French, where morphology substantially influences both word formation and semantic interpretation.

In summary, the primary sources of errors in this research include biases and representativeness of the datasets, inconsistencies in the preprocessing methods, limitations of static embeddings, restrictions of the specific tasks during the evaluation, and the challenge of linguistic diversity. Work on more diverse and representative datasets, contextualized embeddings like BERT or GPT, and the evaluation methodology could address these sources of errors by refining the measures to capture the task-specific nuances and the linguistic intricacies involved.

Conclusion

The paper presents findings and discussions about an overarching comparison of Word2Vec, GloVe, and FastText in three important NLP tasks – namely, NER, sentiment analysis, and POS tagging – both with the use of English and French datasets. In this work, very essential insights about the general applicability of each of these embedding strategies and their multilingual benefits and drawbacks were unveiled.

The results supported the hypothesis of FastText outperforming both Word2Vec and GloVe in the tasks related to morphologically diverse languages, such as French. In all three tasks experimented with, FastText remained superior in the French language, due to its modeling subword capabilities. Indeed, these capabilities allowed FastText to handle complex word structures and out-of-vocabulary terms more successfully than both Word2Vec and GloVe. This brings evidence to the claim that FastText is best suited to applications requiring a deep understanding of morphologically diverse languages.

The hypothesis that GloVe would perform great in sentiment analysis and POS tagging partially held in the English datasets. The dependence of GloVe on global co-occurrence statistics proved to be very handy in the task of sentiment analysis, where the capture of wider semantic relationships is essential. Nevertheless, even though GloVe performed well in POS tagging, FastText's capacity to handle infrequent grammatical structures brought it closer in performance, mainly concerning recall and F1 scores. For NER tasks in English, Word2Vec performed better than both GloVe and FastText. This means that Word2Vec is more capable of modeling contextual relationships, which is a requirement for identifying and classifying entities within a text.

The prediction that FastText would be the most consistent model across tasks and languages was also confirmed. FastText showed consistent performance with minimal variability, which is due to its subword approach, which makes it more adaptable and less reliant on extensive training data. In contrast, GloVe had more variability in low-resource settings, and Word2Vec struggled with OOV words, especially in French, where morphological complexity was a challenge.

In summary, this paper shows that embedding techniques' performance is task- and language-dependent. FastText is the best choice for general-purpose, multilingual NLP applications, especially for morphologically complex languages. GloVe outperforms other techniques in tasks that depend on global semantic understanding, such as sentiment analysis, in high-resource languages like English. Word2Vec remains a good choice for NER-like tasks where contextual sensitivity is important and there is abundant training data.

The results indicate the importance of the embedded methodology selection towards satisfying specific constraints of the desired application. Research could expand in studying integration of

typical word embeddings, along with contextual models or transformer-based structures, beyond static word embeddings for filling gaps that come as limitations with higher effectiveness in poor-resource and other multilingual domains.

References

Baroni, M., Dinu, G., & Kruszewski, G. (2014). Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 238-247.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26, 3111-3119.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., & Mikolov, T. (2018). Learning word vectors for 157 languages. *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*.

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5, 135-146.

Joulin, A., Grave, E., Bojanowski, P., Mikolov, T. (2018). FastText.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Appendix - Datasets

French FQB Dataset: http://alpage.inria.fr/Treebanks/FQB/

Large Movie Review Dataset: https://github.com/SrinidhiRaghavan/AI-Sentiment-Analysis-on-IMDB-Dataset/blob/master/README.md

AlloCine dataset: https://huggingface.co/datasets/tblard/allocine

CoNLL-2003 Dataset: https://www.kaggle.com/datasets/alaakhaled/conll003-englishversion?resource=download&select=valid.txt

Babelscape Dataset (for French): https://huggingface.co/datasets/Babelscape/wikineural