Word Embedding as a Semantic Feature Extraction Technique in Arabic Natural Language Processing: An Overview

Ghizlane Bourahouat
ITQAN Team, LyRICA Laboratory
ESI, Morocco
ghizlane.bourahouat@esi.ac.ma

Manar Abourezq
ITQAN Team, LyRICA Laboratory
ESI, Morocco
mabourezq@esi.ac.ma

Najima Daoudi
ITQAN Team, LyRICA Laboratory
ESI, Morocco
ndaoudi@esi.ac.ma

Abstract: *Feature extraction has transformed the field of Natural Language Processing (NLP) by providing an effective way to represent linguistic features. Various techniques are utilised for feature extraction, such as word embedding. This latter has emerged as a powerful technique for semantic feature extraction in Arabic Natural Language Processing (ANLP). Notably, research on feature extraction in the Arabic language remains relatively limited compared to English. In this paper, we present a review of recent studies focusing on word embedding as a semantic feature extraction technique applied in Arabic NLP. The review primarily includes studies on word embedding techniques applied to the Arabic corpus. We collected and analysed a selection of journal papers published between 2018 and 2023 in this field. Through our analysis, we categorised the different feature extraction techniques, identified the Machine Learning (ML) and/or Deep Learning (DL) algorithms employed, and assessed the performance metrics utilised in these studies. We demonstrate the superiority of word embeddings as a semantic feature representation in ANLP. We compare their performance with other feature extraction techniques, highlighting the ability of word embeddings to capture semantic similarities, detect contextual associations, and facilitate a better understanding of Arabic text. Consequently, this article provides valuable insights into the current state of research in word embedding for Arabic NLP.*

Keywords: *ANLP, feature extraction, word embedding, BERT, transformers.*

Received July 15, 2023; accepted January 30, 2024
https://doi.org/10.34028/iajit/21/2/13

1. Introduction

Natural Language Processing (NLP), a subset within the realm of Artificial Intelligence (AI), focuses on enabling machines to understand and process human language. It involves employing various computational methods to analyse and represent natural language texts at various linguistic levels. The goal is to equip machines with language-processing abilities akin to human capabilities across a broad spectrum of tasks and applications [27]. NLP technologies have paved the way for the creation of a myriad of applications, spanning Sentiment Analysis (SA), speech recognition, Optical Character Recognition (OCR), information retrieval, machine translation, question answering, text summarization, and more.

Given that Arabic is the fifth most widely spoken language worldwide [26] and the fourth most utilised language on the internet [25], there has been a growing interest among researchers in applying NLP techniques to Arabic. However, this endeavour requires additional efforts due to the unique challenges associated with the language. These challenges include morphological complexity, orthographic ambiguity, dialectal variations, orthographic noise, and limited resources for training and evaluating Machine Learning (ML) and Deep Learning (DL) models.

To address these challenges, dedicated efforts have been made to develop customised NLP techniques tailored to the Arabic language, leading to the emergence of Arabic Natural Language Processing (ANLP).

ANLP involves creating methodologies and tools to facilitate the utilisation and analysis of the Arabic language in both written and spoken contexts. The ANLP workflow includes several crucial steps essential for processing natural language in Arabic. Data collection is the initial stage where the required data is gathered. Data annotation follows, wherein additional information related to the targeted task, such as sentiment analysis, is added to the data. Normalization techniques are employed to reduce the diversity of information that needs to be processed by the computer. Tokenization is performed to split the text into individual words while stemming generates morphological variants based on root words. Feature extraction is then applied to convert raw text data into numerical features. The goal of the article lies in accomplishing a specific task, which could involve the classification or prediction of sentiments, emotions or opinions of humans towards products, issues or services.

Although there are numerous studies related to ANLP, we can identify several areas of improvement,

specifically regarding the aspect of feature extraction and its semantic approach known as word embedding.

Word embedding represents words as real-valued vectors in a multi-dimensional space, capturing their semantic and syntactic properties. This facilitates NLP tasks as classification and DL algorithms prefer numerical inputs. By applying the distributional hypothesis, word embeddings achieve impressive results in language understanding [25]. Although widely used in English, word embedding in ANLP is less explored. Thus, this paper focuses on exploring and comparing word embedding techniques in ANLP to improve understanding and performance in Arabic language processing.

The paper is structured as follows. Section 2 provides an overview of the context and existing word embedding techniques in ANLP. Section 3 describes the research methodology and inclusion criteria for the selected papers. Section 4 presents the findings on various word embedding techniques and their performances. Lastly, section 5 discusses the results and provides suggestions for future research directions.

2. General Context

2.1. Feature Extraction

Feature extraction is crucial in ANLP tasks, enabling effective representation and analysis of textual data. It captures linguistic properties, semantic information, and structural patterns, enhancing performance in sentiment analysis, machine translation, text summarisation and more. Extracting meaningful features enables deeper linguistic analysis, improving accuracy and efficiency in ANLP applications.

Feature extraction is the procedure of choosing and transforming raw data into a set of pertinent features that effectively represent and describe the data. In the realm of NLP, feature extraction specifically focuses on extracting meaningful and informative features from text-based data [42]. This process involves extracting words from the text data, which are subsequently converted into features that are utilised by classifiers [12]. By combining variables into components, feature extraction enables the identification of the most valuable features, ultimately reducing the data volume [21]. Various techniques are utilised for feature extraction to transform unstructured textual data into structured representations suitable for machine learning algorithms as illustrated in Figure 1. These techniques include statistical features, such as word frequencies, document lengths, average word lengths, or sentence complexity, and provide information about the distribution and characteristics of the text. Syntax-based features capture syntactic structures and relationships within sentences, including parse trees, dependency graphs, and syntactic paths. Word embeddings generate dense vector representations of words in a continuous vector space, capturing distributional properties. The selection of feature extraction techniques depends on the specific NLP task and the characteristics of the text data being analysed. This step can improve the performance of several tasks such as sentiment analysis, question answering, text generation, text translation and text summarisation [13].

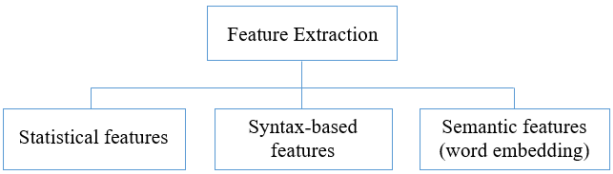

Figure 1. Feature extraction techniques.

2.2. Statistical Features

Statistical features play a crucial role in NLP tasks by capturing various quantitative properties of text documents. These features provide valuable insights into the structural characteristics and composition of textual data [23]. Metrics such as document length, word count, sentence count, and average word length are commonly employed as statistical features and provide insights into the size, vocabulary richness, syntactic complexity, and linguistic characteristics of the text [23]. These features contribute to a comprehensive representation of textual data and play a crucial role in text classification tasks.

2.3. Syntax-based Features

Syntax-based features are essential in capturing grammatical structure and dependencies within sentences. Techniques like dependency parsing, part-of-speech tagging, and chunking extract these features. Dependency parsing identifies relationships between words, Part Of Speech (POS) tagging assigns grammatical tags, and chunking groups words into syntactic units. Incorporating these features enhances classification models by leveraging structural and grammatical properties, improving understanding of linguistic patterns [32].

2.4. Word Embedding

Word embedding, as a semantic approach to feature extraction, captures the underlying meaning and semantic relationships between words, enabling the representation of textual data in a dense vector space. This representation enhances the understanding of semantic similarities and contextual associations, facilitating more effective analysis and interpretation of text.

There is a wide range of state-of-the-art approaches that have been created to tackle the task of feature extraction and specifically word embedding. In this subsection, we give an overview of the most used ones in the Arabic language. These methods could be categorised as being either static or contextualised.

2.4.1. Static Word Embedding

Words are represented as a fixed dense vector in static word embedding approaches, where each different word is assigned only one pre-computed embedding. In ANLP, there are various static word embedding techniques that are used, namely Word2Vec, FastText, AraVec, Sense2Vec, Glove, ArWordVec and MUSE. We will present, in this subsection, each one of these word embedding models [19].

2.4.1.1. Word2Vec

The Word2Vec model, introduced by Mikolov [39], is a neural network architecture comprising an input layer, an output layer, and a hidden layer. The hidden layer, devoid of activation functions, has several neurons equal to the dimension of the word's vector representation in the word embedding. Word2Vec addresses two limitations of the "one-hot" representation: the lack of syntactic and semantic relationships among word vectors and the inefficient utilization of sparse space. By training on large datasets, the Word2Vec model effectively captures the semantics and syntax of words, enabling accurate measurement of word similarity.

Word2Vec consists of two approaches which are the Continuous Bag-Of-Words Model (CBOW) and the Skip-gram model:

- CBOW: the CBOW approach employs a log-linear classifier to classify the middle word based on the surrounding words from both the future and history [13].
- Skip-gram (SG): both the CBOW and Skip-gram approaches of the Word2Vec model share a similar structure with one key difference. In CBOW, the input to the neural network is the context words, and the output is the middle word. In contrast, the Skip-gram model takes the current word as input and predicts the surrounding context words.

2.4.1.2. FastText

FastText utilizes a pre-trained word embedding layer developed by Facebook specifically for the Arabic dataset, which comprises a vocabulary size of 2 million and a vector dimension of 300 [30]. The fundamental idea behind FastText's embedding layer is to generate word vectors based on semantic meaning. It achieves this by utilizing n-grams from input sentences and appending them to the end of phrases. FastText can generate word vectors for unknown or out-of-dictionary words by taking into account the morphological features of words. Even if a word was not observed during training, its embedding can be determined by breaking it down into n-grams. FastText has also two methods, which are CBOW and SG.

2.4.1.3. AraVec

AraVec is a pre-trained distributed word representation model designed specifically for the Arabic language. Its purpose is to offer the ANLP research community freely accessible and powerful word embedding models. AraVec was constructed using diverse Arabic text resources to ensure broad domain coverage [4, 15]. Similar to Word2Vec and FastText, AraVec supports two architectures: CBOW and SG. It was created using the Word2Vec SG technique and trained on web pages containing Arabic content, with each word represented in a 300-dimensional vector space.

2.4.1.4. Glove

GloVe, which stands for Global Vectors for Word representation, is a model based on global corpus statistics and utilizes a weighted least-squares objective with a global log-bilinear regression approach [10, 41]. The underlying principle of this model is that the ratio of word-word co-occurrence probabilities captures the semantic meaning of words. GloVe learns word representations by comparing the co-occurrence probability of two words, *i* and *j*, with various probe words *k* in Equation (1). This ratio is significant when the context word is associated with *i*, small when it is associated with *j*, and close to one when the context word is related to both words.

$$\frac{P_{ik}}{P_{jk}} = F\left(w_i, w_j, \widetilde{w}_k\right) \tag{1}$$

2.4.1.5. Sense2Vec

Sense2Vec is an extended version of the Word2Vec algorithm that generates vector space representations for words based on large corpora. Unlike Word2Vec, Sense2Vec creates embeddings for "senses" rather than individual word tokens. A sense in Sense2Vec refers to a word combined with a label that represents the contextual information in which the word is used. These labels can include POS tags, polarity indicators, entity names, and dependency tags, among others. By incorporating these labels, Sense2Vec captures more nuanced information about word usage and context [19].

2.4.1.6. MUSE

Multilingual Unsupervised and Supervised Embeddings (MUSE) is a Python library designed to facilitate the development and evaluation of cross-lingual word embedding and NLP applications. It offers a unique approach by leveraging multilingual word embeddings instead of relying on language-specific training or translations for text classification tasks. By training models across multiple languages, MUSE enables developers to scale their NLP projects and achieve faster and more efficient results [44].

2.4.1.7. ArWordVec

ArWordVec is a collection of pre-trained word embedding models specifically created for Arabic NLP research. These models are derived from a large database of Arabic tweets that cover a wide range of topics, including education, healthcare, politics, and social affairs. By incorporating diverse usage of words within different domains and hashtags, ArWordVec aims to enhance the effectiveness of word embeddings in capturing the nuances and context of Arabic text, particularly in the context of Twitter data. These pre-trained models can be leveraged by researchers to facilitate their Arabic NLP projects and analysis [29].

2.4.2. Contextual Word Embedding

In this class, models work on the concept of contextual string embeddings. Word embeddings are contextualised by the words that surround them. As a result, depending on the surrounding text, it generates multiple embeddings for the same word. In ANLP, there are numerous contextualised word embedding techniques that were used, namely AraBERT, MARBERT, QARIB, ALBERT, XLM, mBERT, CaMelBERT, AraELECTRA and Flair.

We'll start by introducing the various pre-trained Arabic based on the BERT model as shown in Figure 2. BERT is a text representation technique combining a variety of state-of-the-art DL algorithms. It can be used for the tokenisation, embedding and other tasks such as classification.

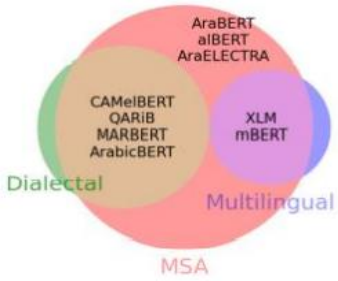

Figure 2. Multiple Arabic BERT pre-trained models [24].

2.4.2.1. AraBERT

AraBERT is a multi-layer bidirectional transformer model introduced in reference [20]. The primary concept behind AraBERT involves pre-training deep bidirectional representations using unlabeled text, considering context from both preceding and succeeding directions. Subsequently, all the parameters of the model are fine-tuned on a specific downstream task. AraBERT relies on two key processes during pre-training: masked language modelling and next-sentence prediction. These processes contribute to the model's ability to understand and generate contextual representations of Arabic text, enabling it to perform effectively on various NLP tasks.

2.4.2.2. MARBERT

MARBERT is an extensively pre-trained masked model developed on a vast collection of datasets comprising Modern Standard Arabic (MSA) and Dialectal Arabic (DA) [18]. This model was specifically designed to excel in downstream tasks involving dialectal Arabic. To train MARBERT, approximately 1 billion Arabic tweets were randomly chosen from a substantial in-house dataset encompassing around 6 billion tweets. By leveraging this large and diverse data source, MARBERT aims to capture the intricacies and nuances of Arabic language usage, enabling it to deliver strong performance on various NLP applications.

2.4.2.3. mBERT

Multilingual BERT is a language model that has been trained on a corpus of 104 languages [24]. It serves as a universal language model, capable of supporting over 100 languages, including Arabic, Dutch, French, German, Italian, and Portuguese. The training of the model encompasses various domains, such as social media posts and newspaper articles, ensuring a wide coverage of language usage across different contexts. With its extensive multilingual training, Multilingual BERT offers a versatile tool for natural language processing tasks in diverse languages and domains.

2.4.2.4. QARIB

QCRI Arabic and Dialectal BERT (QARIB) is a specialized dialectal BERT model that has been trained on a vast dataset consisting of 420 million tweets and 180 million sentences of text. The training corpus comprises a total of 14 billion tokens, and the model employs a vocabulary size of 64,000 with 12 layers. The data for the tweets was collected using the Twitter API, ensuring a diverse range of dialectal Arabic language usage. QARIB offers enhanced capabilities for processing and understanding dialectal Arabic text, making it a valuable resource for various natural language processing tasks in this linguistic [1].

2.4.2.5. CAMeLBERT

CAMeLBERT is a comprehensive suite of BERT models that have been pre-trained on Arabic texts, encompassing various sizes and variants. It includes pre-trained language models specifically designed for Modern Standard Arabic (MSA), Dialectal Arabic (DA), and Classical Arabic (CA). Additionally, there is a model pre-trained on a combination of all three variants. Furthermore, CAMeLBERT offers additional models that have been pre-trained on a scaled-down subset of the MSA variant, providing a range of options to suit different Arabic language processing requirements [8].

2.4.2.6. ALBERT

A Lite BERT (ALBERT) is an optimized version of BERT that aims to improve its performance while reducing its computational requirements. The largest ALBERT model, known as ALBERT-xxlarge, has approximately 70% of the parameters of BERT-large [24]. Despite having fewer parameters, ALBERT achieves significant improvements in performance across various NLP tasks. ALBERT follows a Transformer-based neural network architecture and incorporates two parameter reduction strategies. These strategies help enhance training efficiency and reduce memory usage compared to the original BERT model.

2.4.2.7. XLM

XLM is a transformer based architecture that is pre-trained using one of the following language modelling objectives [8]:

- Causal Language Modelling: to model the probability of a word given the previous words in a sentence.
- Masked Language Modelling: the masked language modelling objective of BERT.
- Translation Language Modelling: a translation language modelling objective for improving cross-lingual pre-training.

2.4.2.8. AraELECTRA

Arabic ELECTRA is a question-answering system for Arabic Wikipedia that utilises a language representation model. The underlying model, AraELECTRA, has been pre-trained on a sizable corpus of (MSA) data, employing the RTD objective. With a total of 136 million parameters, AraELECTRA is built as a bidirectional transformer encoder model. It consists of 12 encoder layers, 12 attention heads, a hidden size of 768, and a maximum input sequence length of 512. The system is powered by streamlit, a framework for building interactive web applications [3].

2.4.2.9. Flair

Flair is an advanced NLP framework that is open-source and developed by Zalando Research, a division of the fashion platform Zalando. The goal of Zalando Research is to apply experimental approaches and theory to scale technology in the fashion industry. Flair, released in July 2018, introduces a unique method of leveraging natural language modelling to acquire contextualised representations of human language from extensive corpora. These representations contain rich semantic and syntactic information that can directly enhance various downstream NLP tasks.

As we can see, several word embedding models can be used in the Arabic context, but there is still a need to detect the most efficient and powerful ones in the ANLP tasks. Thus, our article aims to find the most used and useful embedding techniques in ANLP. We'll also investigate the implemented ML and/or DL model with the embedding models. To do so, we adopted the research methodology presented in the next section.

3. Methodology

This paper presents a literature review of recent studies in the field of ANLP. The goal is to identify the most commonly used word embedding techniques, as well as the issues they confront. We examined articles that were published between 2018 and 2023 in journals that are indexed on the Web of Science and Scopus databases to find relevant studies. Articles were chosen based on their detailed content and rigorous peer review. They were identified using the keywords "Arabic natural language processing", "ANLP and word embedding techniques," "Arabic natural language processing and word embedding", "feature extraction", "Arabic Sentiment Analysis and word embedding", and "Arabic Sentiment Analysis and feature extraction".

- **Research Questions**

The following research questions were included as a guide to frame the current review:

- Q1. What kind of feature extraction techniques are mostly used in ANLP tasks?
- Q2. What kind of NLP tasks are used for evaluating the performance of the Word embedding?
- Q3. What are the ML/DL algorithms used with Arabic word embedding techniques?

After retrieving research articles based on the search parameters, we initially screened each paper by reviewing its title, abstract, conclusion, and keywords. Following this preselection process, we proceeded to the second stage of evaluation, where we thoroughly examined the entire content of the selected papers. During this stage, we conducted in-depth research and analysis on the articles that met our selection criteria.

- **Inclusion Criteria**
  - Word embedding with the Arabic language is discussed in these articles.
  - Articles discussing feature extraction in the ANLP context.
  - Articles discussing word embedding in ANLP context.
  - Articles describing the proposed ANLP technique in depth and evaluating it.
  - Articles published in high-impact journals between 2018 and 2023.

- **Exclusion Criteria**

Articles in which the techniques or evaluation methods are not clearly explained.

- Articles in which the content is NLP applied to Arabic with other languages.

- Articles that are systematic literature reviews.
- Articles that were not written in English.

Initially, approximately 80 papers were retrieved, and the inclusion and exclusion criteria were subsequently applied to them. Following this screening process, 40 papers were identified as relevant and met the established criteria. These selected papers were then subjected to in-depth analysis for this study. Figure 3 provides a detailed depiction of the selection process.

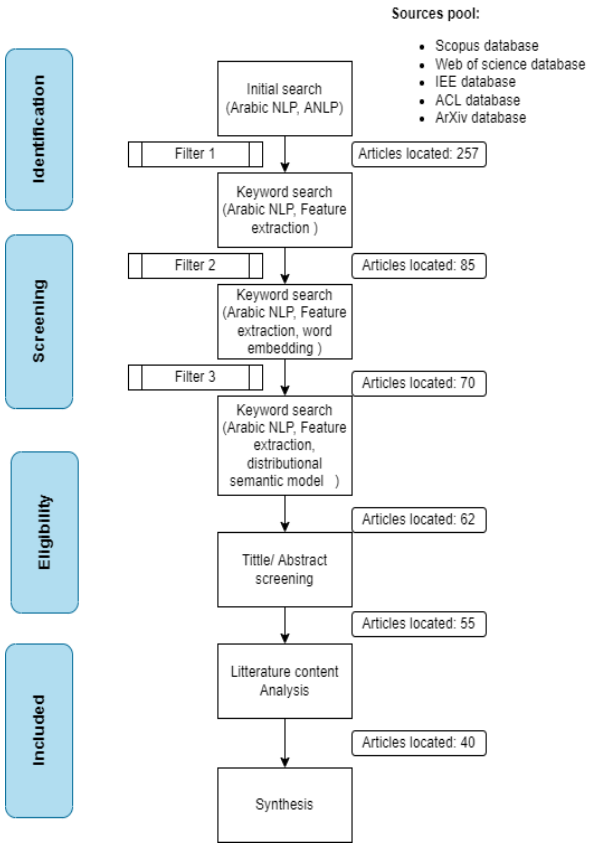

Figure 3.The process of searching.

As mentioned above, out of the 257 articles, 40 treated the word embedding concept, which is more than 50% of the returned results. We explore these articles more in detail in the next section. In Figure 4, the most dominant source is journal articles, which constitute almost 70% of the total sources. However, conference articles are 30%.

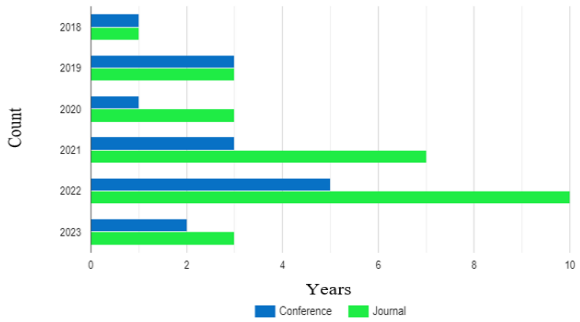

Figure 4. Type of searched studies.

Given that the search strategy has a direct influence on the relevance and comprehensiveness of the retrieved studies, we successfully utilised the prominent databases illustrated in Figure 5, namely Scopus, IEEE, Springer, ScienceDirect, and ACL Aclanthology.

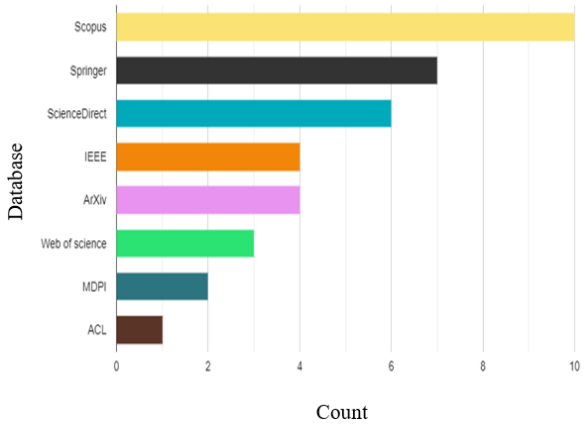

Figure 5. Searched studies across databases.

Among these databases, Scopus yielded the highest number of papers, followed by Springer, ScienceDirect, and IEEE.

4. Results and Analysis

Starting with the static Word embedding. The Continuous Bag-Of-Words approach was employed by [2, 7, 24, 25, 27]. The Skip-gram approach was also investigated in [2]. FastText was used in [11, 28, 29, 30]. The Glove model was implemented by [2, 32]. As for the sens2vec model, it was employed in [7]. The MUSE model was used by [33]. ArWordVec was implemented in [35] as shown in the table below. Like AraVec and FastText, both approaches CBOW and SG were investigated in [35].

A range of embedding techniques were employed in conjunction with both ML and DL algorithms. Among the ML algorithms, the commonly utilised ones were Support Vector Machine (SVM), Logistic Regression (LR), Random Forest (RF), Naïve Bayes (NB), Stochastic Gradient Descent (SGD), and others, as specified in detail in Table 1. On the other hand, for DL models, the prevalent choices included Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), Bidirectional LSTM (Bi-LSTM), and Gated Recurrent Units (GRU).

As the evaluation and comparison between the models implemented with different algorithms and data is subject to some confusion, we have referred to several criteria which are the used algorithm (ML, DL), the type of preliminary task, the metric used to measure the performance and finally the performance of the model. Table 1 presents more details.

Table 1. They studied static word embedding techniques.

Word embedding technique	Articles	Objective Classification	Algorithm	Metric	Performance
Word2Vec	[4]	Multi-class	Att-GRU	F1 score	97,82%
			Bi-GRU		97,23%
			Bi-LSTM		97,15%
			Att-LSTM		97,15%
			CNN		96,71%
			CNN-GRU		96,68%
			CNN-LSTM		94,56%
Word2ec-CBOW	[13]	Multi-class	CNN	Accuracy	99,82%
		Binary	Nu SVC	F1 score	93,48%
			RF, SVM, SGD		91,22%
			CV, SGD, SVM		90,94%
			RF		89,63%
			LR		89,59%
			LR, RF, SGD		89,26%
			Linear SVC		84,61%
			SGD		77,20%
			LR, RF, BNB		75,48%
			BNB		65,19%
	[24]	Multi-class	CNN	F1 score	96,7%
	[2]	Binary	CNN-LSTM	Accuracy	93,58%
	[4]	Binary	SGD	F1 score	88%
			LSVC		85%
			LR		85%
			GNB		83%
			CNN		82%
			Bi-LSTM		80%
			RF		80%
			LSTM		80%
	[17]	Binary	LSTM	Accuracy	81,31%
			RCNN		78,46%
			CNN		75,72%
	[27]	Multi-class	NB	Accuracy	46%
			MLP		36%
			SVM		24%
Word2ec- SG	[24]	Multi-class	CNN	F1 score	96,8%
	[2]	Binary	CNN-LSTM	Accuracy	96,68%
FastText	[29]	Binary	NuSVC		86,74%
			LR		84,53%
			Linear SVC		84,20%
			RF	Accuracy	83,65%
			SGD		81,88%
			Gaussian NB		70,94%
	[13]	Binary	CNN	F1 score	62,92%
		Multi-class	CNN		62,60%
			CNN		34,47%
FasText- CBOW	[2]	Binary	CNN-LSTM	Accuracy	93,79%
	[35]	Multi-class	RNN-LSTM	Accuracy	91%
	[6]	Binary	CNN-LSTM	Accuracy	88,90%
	[36]	Binary	Bi-LSTM	F1 score	88,26%
			CNN		86%
	[4]	Binary	CNN	F1 score	80%
			LSTM		79%
			Bi-LSTM		79%
FasText- SG	[2]	Binary	CNN-LSTM	Accuracy	96,10%
	[31]	Binary	CNN-LSTM	Accuracy	90,75%
	[28]	Binary	LSV	Accuracy	89%
	[9, 45]	Multi-class	NuSVC		84.89%
			Random Forest		82.89%
			Logistic Regression		82.80%
			Linear SVC		81.26%
			SGD		76.97%
			Gaussian NB	Precision	67.60%
AraVec	[11]	Binary	NuSVC		87,51%
			RF		85,86%
			LR		83,98%
			Linear SVC	Accuracy	83,76%
			SGD		81,44%
			Gaussian NB		75,25%
	[37]	Binary	CNN	F1 score	73,20%
			CNN		71%
		Multi-class	CNN		51,03%
	[12]	Multi-class	CNN	F1 score	53,4%
	[16]	Binary	CNN	F1 score	48,6%
AraVec-CBOW	[39]	Multi-class	CNN	Accuracy	89,19%
		Binary	CNN		83,53%
	[36]	Binary	Bi-LSTM	F1 score	86,66%
			CNN		86%
AraVec- SG	[36]	Binary	BLSTM	F1 score	89%
			CNN		88,01%
	[14]	Multi-class	CNN	Accuracy	86%
		Binary	CNN		84,34%
	[21]	Binary	LSV	Accuracy	86%
	[31]	Binary	CNN-LSTM	Accuracy	81,35%
	[38]	Multi-class	CNN	F1 score	53,4%
		Binary	CNN		48,6%
	[37]	Multi-class	CNN	F1 score	51,03%
Glove	[2]	Binary	CNN	Accuracy	94,80%
	[5]	Binary	LSTM	Accuracy	89,82%
	[30]	Multi-class	NuSVC		82.69%
			RF		79.50%
			LR		78.40%
			Linear SVC	Precision	77.12%
			SGD		71.61%
			GNB		65.22%
Sens2vec	[7]	Binary	LRCV		89,4%
			SGD		89,3%
			NuSVC		89,2%
			Linear SVC	Accuracy	88,8%
			RF		86,1%
			Gaussian NB		77,8%
MUSE	[33]	Binary	CNN		70%
		Multi-class	CNN		65%
			Bi-LSTM -CNN	Accuracy	60%
ArWordVec - CBOW	[29]	Binary	Bi-LSTM	F1 score	88,44%
			CNN		84,38%
ArWordVec - SG	[35]	Binary	Bi-LSTM	F1 score	88,56%
			CNN		84,33%

Moving to the contextualised embedding techniques. From the reviewed articles, we found that the CAMeLBERT model was used by [8]. QARIB model was employed in [11, 28, 34]. Multilingual BERT was used in, [5, 13, 24]. As for the MARBERT model, it was employed in [13, 24, 38, 41]. Furthermore, the AraBERT model was investigated by [13, 24, 33, 37, 42, 43] as mentioned in the Table 2. It is worth mentioning that most of the Arabic pre-trained BERT models are used to perform the whole process of ANLP. Therefore, we found that they were generally used to generate the embeddings but also as classifiers, except in [5, 33] where the SVM algorithm was used. These results are detailed in Table 2.

Table 2. The studied contextualised word embedding techniques.

Word embedding technique	Article	Objective Classification	Algorithm	Metric	Performance
AraBERT	[23]	Binary	AraBERT	Accuracy	96,2 %
	[13]	Multi-class	AraBERT	Accuracy	93,8%
	[42]	Multi-class	AraBERT	Accuracy	92,6%
	[33]	Multi-class	AraBERT	Precision	90%
	[33]	Multi-class	SVM	Precision	87%
	[24]	Multi-class	AraBERT		89,6%
				Accuracy	75,5%
	[43]	Binary	AraBERT	F1 score	80%
		Multi-class			64,3%
					62,5%
QARIB	[24]	Multi-class	QARIB	Accuracy	95,4%
					93,3%
					82,3%
	[31]	Multi-class	SVM	Precision	90%
	[28]	Binary	QARIB	F1 score	79,1%
		Multi-class			67,1%
					63,1%
mBERT	[13]	Binary	mBERT	Accuracy	95,7%
	[24]	Multi-class	mBERT	Accuracy	93,8%
					85,5%
					74,3%
	[5]	Multi-class	SVM	Precision	85%
CAMeLBERT	[24]	Multi-class	CAMeLBERT	Accuracy	95,6%
					92,5%
					83,3%
MARBERT	[24]	Multi-class	MARBERT	Accuracy	95,2%
					93,2%
			MARBERT		81,9%
	[41]	Multi-class	MARBERT	Pearson's correlation	86,1
					70,9
	[13]	Binary	MARBERT	F1 score	74,80%
					60%
	[38]	Multi-class	MARBERT	F1 score	70,2%
					60,9%
ALBERT	[24]	Multi-class	ALBERT	Accuracy	94,5%
					89,6%
					78,3%
XLM	[24]	Multi-class	XLM	Accuracy	95,1%
					89,3%
					77,1%
AraELECTRA	[8]	Multi-class	AraELECTRA	Accuracy	95,2%
					90,6%
					79,7%
Flair	[37]	Binary	CNN	F1 score	57,31%
		Multi-class	CNN		39%
		Binary	CNN		37,16%

Based on the sample of studied articles, it seems that static word embeddings are the dominant choice in the field of ANLP, accounting for approximately 62% of usage. Following closely behind are contextualised word embeddings, which make up approximately 38% of the usage in this context as shown in Figure 6.

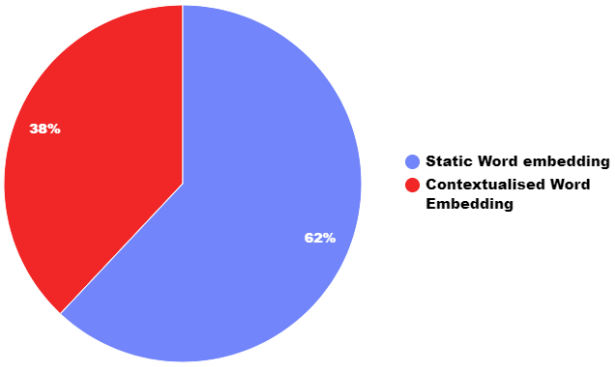

Figure 6. Distribution of Word embedding's categories.

Figure 7 illustrates the diverse range of embedding approaches employed in the analysed articles.

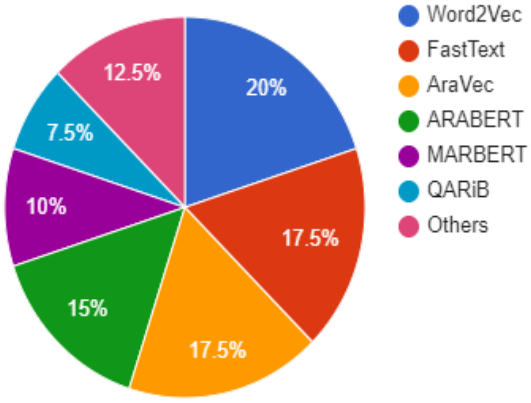

Figure 7. Distribution of Word embedding techniques used.

According to our observations, it appears that the Word2Vec model was the most commonly employed for embedding, accounting for 20% of the sample. Following closely, FasText and AraVec were utilised at a rate of 17.5%. In the third position, the AraBERT model constituted 15% of the embeddings used. Moreover, MARBERT and QARIB were employed in 10% and 7.5% of the cases, respectively. The last position refers to the less commonly used models, which include Glove, CAMelBERT, ALBERT, XLM, AraELECTRA, Flair, Sense2Vec, MUSE and ArWordVec as exposed in Figure 7.

In comparing statistical, syntax-based, and semantic feature extraction techniques for ANLP, we observe distinct characteristics. Statistical approaches, driven by frequency and distributional patterns, excel in capturing surface-level information. Syntax-based techniques, leveraging grammatical structures, offer insights into sentence organization and syntactic relationships. On the other hand, semantic feature extraction, exemplified by word embeddings, goes beyond surface patterns and syntax, capturing contextual and semantic relationships between words. While statistical and syntax-based methods are effective for tasks emphasizing frequency and syntactic structures, semantic feature extraction proves essential for tasks requiring a deeper understanding of context and meaning. In our focus on semantic feature extraction, particularly through word

embeddings, we prioritize a nuanced representation of semantics that aligns with the intricacies of the Arabic language. This choice is justified by the growing importance of semantic understanding in various ANLP applications, including sentiment analysis, machine translation, and information retrieval, where a richer understanding of context is paramount.

In the scope of this article, our primary focus is on delving into the intricacies of word embeddings within natural language processing. It is crucial to emphasize that our emphasis on this specific technique does not imply a neglect of the impact of other techniques. Rather, it reflects a targeted exploration aimed at providing a comprehensive understanding of the word embedding aspect. We acknowledge the diversity of techniques within the broader landscape of natural language processing, each contributing uniquely to the performance across various applications and domains. While we spotlight word embeddings in this discussion, it is important to recognize that the effectiveness of any technique can be context-dependent. The varied interplay of techniques in different linguistic scenarios underscores the dynamic nature of natural language processing, and we encourage further exploration into the collective impact of these methodologies across diverse applications and domains.

5. Discussion

The objectives of the articles included in the study varied depending on the motivations of the researchers and the expressed needs. As presented in Figure 8, a multitude of objectives were targeted, with the most recurring one being Arabic Sentiment Analysis (ASA), accounting for 63% of the articles. Emotion detection and sarcasm and irony detection followed with equal percentages of 11.1% each. Most of the reviewed articles focused on ASA, although other tasks such as emotion detection, sarcasm detection, hate speech detection, text generation, and text summarization were also addressed. Moreover, during the examination of these articles, we identified three main tasks performed: binary classification, ternary classification, and multi-class classification.

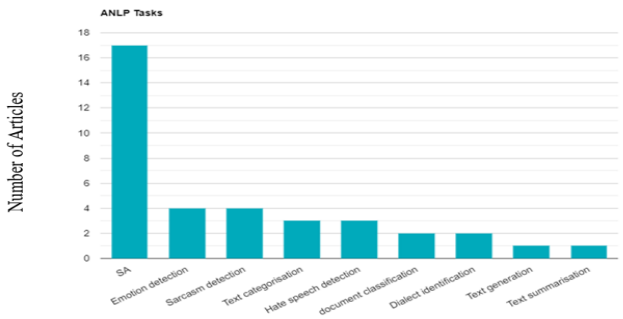

Figure 8. Distribution of articles' objectives.

Another important aspect is the dialect used in each study. In the sample studied, it was found that MSA was the most used language implementation as depicted in Figure 9, accounting for 47.1% of the cases. The Egyptian dialect and Algerian dialect were the next most frequently utilised, representing 11.8% and 9.8% respectively.

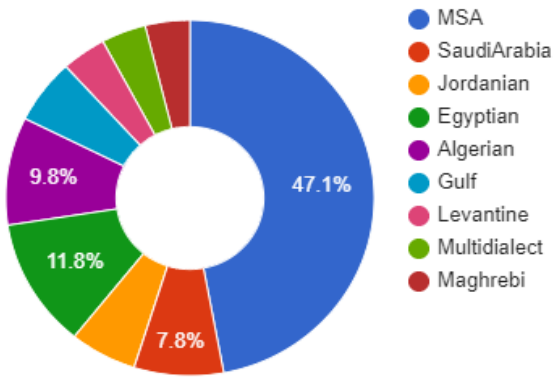

Figure 9. Distribution of Arabic dialects.

As for the performance of the word embedding techniques, we notice various performances. The latter was related to the word embedding technique used and the ML or DL model implemented. To have a better observation, we proposed Table 3, presenting all the overviewed models and techniques with the best result performance.

Table 3. Word embedding techniques in ANLP with their best performance.

Word Embedding	Articles	Best Performance
Word2Vec	[2,13, 24, 27, 31, 34]	SG: 96,8% (F1 score)
		CBOW: 91,22% (F1 score)
FastText	[2, 4, 9, 28, 29, 31, 34]	SG: 96.10% (Accuracy)
		CBOW: 93,79% (Accuracy)
AraVec	[16, 22, 31, 34, 37, 39]	89,19% (F1 score)
Glove	[2, 9, 40]	94,80% (Accuracy)
MUSE	[31]	70% (Accuracy)
ArWordVec	[29]	88,56%(F1 score)
Sense2Vec	[26]	89,4% (Accuracy)
AraBERT	[13, 24, 33, 37, 43]	93,8% (Accuracy)
QARIB	[24, 28, 31]	95,4% (Accuracy)
MARBERT	[13, 24, 38, 41]	95,2% (Accuracy)
mBERT	[13, 24, 40]	95,7% (Accuracy)
CAMeLBERT	[24]	95,6% (Accuracy)
XLM	[24]	95,1% (Accuracy)
AraELECTRA	[24]	95,2% (Accuracy)
ALBERT	[24]	94,5% (Accuracy)
Flair	[37]	54.15% (Accuracy)

From the resulting performance, we find that the best combination for the Word2Vec model is the SG variant with the CNN model as found in [24]. The SG variant had also the highest performance when combined with CNN-LSTM as a classifier [2]. In [24], the Word2Vec technique combined with the Att-GRU classifier led to an F1 score of 97.96%. The FastText-SG was also the best compared to other approaches that use SG, with a performance of 96.1% when combined with the CNN-LSTM algorithm [2]. In [24], we found that the accuracy resulted from the BERT pre-trained models, namely AraBERT, CAMeLBERT, MARBERT, QARIB,

mBERT, AraELECTRA, ALBERT, XLM, varies between 93.8% and 95.6%. In [26], the Sense2Vec model resulted in an accuracy of 89.4% based on the concept of senses which is, by default, included in the Arabic Bert pre-trained models. The Glove model achieved 94.8% together with the CNN as seen in [2]. The AraVec model achieved its highest performance when combined with the CNN model, achieving an accuracy rate of 89.19% [39].

For the static category, we find that the AraVec model, which was based on Word2Vec architecture to get adapted for Arabic, has also achieved a high accuracy which has reached 89.19%. ArWordVec is another Arabic word embedding model that utilizes two techniques, CBOW and Skip-gram. This model demonstrated an accuracy of 88.56% in the study. It is worth noting that the presented models were meticulously constructed, incorporating multiple Arabic text resources to ensure comprehensive coverage across various domains.

A general observation from the study indicates that contextualised word embedding models tend to outperform static ones. Notably, Arabic pre-trained BERT models have shown even better results. This finding is reasonable since contextualised models consider the word's context rather than solely focusing on its syntax. By capturing contextual information, these models can better understand the nuanced meaning and improve performance in various natural language processing tasks. We also notice that AraBERT, QARIB, MARBERT and mBERT perform better with an accuracy in the range of 93% and 96%.

In comparing static word embeddings, exemplified by Word2Vec, with contextualized word embeddings, represented by models like BERT, distinct advantages and limitations emerge. Static embeddings offer efficiency and interpretability but lack context sensitivity and struggle with polysemy. In contrast, contextualized embeddings excel in capturing context nuances and handling polysemy, though with higher computational demands and increased model complexity. In the Arab context, with its morphological complexity and dialectal variations, the choice between static and contextualized embeddings should be task-driven. Tasks requiring nuanced context understanding may benefit from contextual embeddings, while resource-efficient static embeddings could suit simpler tasks. A judicious combination of both types may offer a balanced approach, harnessing the strengths of each for comprehensive word representation in Arabic natural language processing.

We notice that several ML and DL were implemented in the context of ANLP, resulting in different performances. Although the same ML or DL may be used in two different works, the result will be different due to the difference in the level of the previous phases related to the pre-processing. Therefore, we can say that the model's performance does not depend on a single parameter but on a combination of parameters. The nature of the dataset is important, the following pre-processing steps are crucial (stemming, tokenization, normalization), the embedding of the model is critical and so is the used algorithm.

6. Conclusions

NLP is a complex technique whose aim is to comprehend and understand humans' written and spoken text. It is conducted by following multiple steps starting with the data collection and arriving at the task performed. This technique has to deal with the different challenges that are specific to each language, including Arabic. Indeed, the Arabic language is characterised by a set of specifics, which present challenges in the context of NLP, especially regarding word embedding. For this reason, this paper presents the recent advances in word embedding techniques to study the impact of embedding on the final performance of ANLP tasks. We started by detecting the used embedding techniques in the Arabic context. We conducted a study to extract the most used and useful techniques in this approach and then presented the overall performances. We've also made a synthesis of all considered models together with the implemented ML and/or DL model, the type of NLP task, the calculated metric, and the performance to conclude that the model's performance is determined by a combination of parameters rather than a single parameter. The results showed that the embedding step in ANLP has a high impact on the model's performance. We've also noticed that the Arabic pre-trained BERT achieve the best performances when used to generate the embeddings. Thus, we intend in our future works to focus on the implementation of different word embedding techniques for ANLP, especially the Arabic-BERT pre-trained ones, and try to improve the model's performance.

References

[1] Abdelali A., Durrani N., Dalvi F., and Sajjad H., "Interpreting Arabic Transformer Models," *arXiv Preprint*, vol. arXiv:2201.07434v1, 2022. https://www.researchgate.net/publication/357952504_Interpreting_Arabic_Transformer_Models

[2] Abdelali A., Hassan S., Mubarak H., Darwish K., and Samih Y., "Pre-Training BERT on Arabic Tweets: Practical Considerations," *arXiv Preprint*, vol. arXiv:2102.10684v1, 2021. https://arxiv.org/abs/2102.10684

[3] Abdul-Mageed M., Elmadany A., and Nagoudi E., "ARBERT and MARBERT: Deep Bidirectional Transformers for Arabic," *in Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, Bangkok, pp. 7088-7105,

2021. doi:10.18653/v1/2021.acl-long.551

[4] Alayba A. and Palade V., "Leveraging Arabic Sentiment Classification Using an Enhanced CNN-LSTM Approach and Effective Arabic Text Preparation," *Journal of King Saud University-Computer and Information Sciences*, vol. 34. no. 10, 2021. doi: 10.1016/j.jksuci.2021.12.004

[5] Alharbi A. and Lee M., "Multi-task Learning Using a Combination of Contextualised and Static Word Embeddings for Arabic Sarcasm Detection and Sentiment Analysis," *in Proceedings of the 6th Arabic Natural Language Processing Workshop*, Kyiv, pp. 318-322, 2021. https://aclanthology.org/2021.wanlp-1.39

[6] Al-Hashedi A., Al-Fuhaidi B., Mohsen A., Ali Y., and Al-Kaf H., "Ensemble Classifiers for Arabic Sentiment Analysis of Social Network (Twitter Data) towards COVID-19-Related Conspiracy Theories," *Applied Computational Intelligence Soft Computing*, vol. 2022, 2022. doi: 10.1155/2022/6614730.

[7] Almuzaini H. and Azmi A., "Impact of Stemming and Word Embedding on Deep Learning-Based Arabic Text Categorization," *IEEE Access*, vol. 8, pp. 127913-127928, 2020. doi:10.1109/ACCESS.2020.3009217

[8] Antoun W., Baly F., and Hajj H., "AraELECTRA: Pre-Training Text Discriminators for Arabic Language Understanding," *arXiv Preprint,* arXiv:2012.15516v2, 2020. http://arxiv.org/abs/2012.15516

[9] Ashi M., Siddiqui M., and Farrukh N., "Pre-Trained Word Embeddings for Arabic Aspect-Based Sentiment Analysis of Airline Tweets," *in Proceedings of the International Conference of Advanced Intelligent Systems and Informatics*, Cairo, pp. 7-9, 2020. https://doi.org/10.1007/978-3-319-99010-1_22

[10] Chaimae A., Rybinski M., Yacine E., and Montes J., "Comparative Study of Arabic Word Embeddings: Evaluation and Application," *International Journal Computer Information System and Industrial Management Applications*, vol. 12, pp. 349-362, 2020. https://www.mirlabs.org/ijcisim/regular_papers_2020/IJCISIM_31.pdf

[11] Chouikhi H., Chniter H., and Jarray F., "Arabic Sentiment Analysis Using BERT Model," *Communications in Computer and Information Science*, vol. 1463, pp. 621-632, 2021. doi: 10.1007/978-3-030-88113-9_50

[12] Chowdhury S., Abdelali A., Darwish K., Soon-Gyo J., Salminen J., and Jansen B., "Improving Arabic Text Categorization Using Transformer Training Diversification," *in Proceedings of the 5th Arabic Natural Language Processing Workshop*, Barcelona, pp. 226-236, 2020. https://www.aclweb.org/anthology/2020.wanlp-1.21

[13] Darwish K., Habash N., Abbas M., Al-Khalifa H., and Al-Natsheh H., "A Panoramic Survey of Natural Language Processing in the Arab World," *Communications of ACM*, vol. 64, no. 4, pp. 72-81, 2021. doi: 10.1145/3447735.

[14] El Mahdaouy A., Gaussier E., and El Alaoui S., "Arabic Text Classification Based on Word and Document Embeddings," *in Proceedings of the International Conference on Advanced Intelligent Systems and Informatics*, Cairo, pp. 32-41, 2017. doi: 0.1007/978-3-319-48308-5_4

[15] El Moubtahij H., Abdelali H., and Tazi E., "AraBERT Transformer Model for Arabic Comments and Reviews Analysis," *International Journal of Artificial Intelligence*, vol. 11, no. 1, pp. 379-387, 2022. doi: 10.11591/ijai.v11.i1.pp379-387

[16] Elfaik H. and Nfaoui E., "Combining Context-Aware Embeddings and an Attentional Deep Learning Model for Arabic Affect Analysis on Twitter," *IEEE Access*, vol. 9, pp. 111214-111230, 2021. doi: 10.1109/ACCESS.2021.3102087

[17] Fawzy M., Fakhr M., and Rizka M., "Word Embeddings and Neural Network Architectures for Arabic Sentiment Analysis," *in Proceedings of the 16th International Computer Engineering Conference*, Cairo, pp. 92-96, 2020, doi:10.1109/ICENCO49778.2020.9357377.

[18] Fouad A., Mahany A., Aljohani N., Abbasi R., and Hassan S., "Ar-WordVec: Efficient Word Embedding Models for Arabic Tweets," *Soft Computing*, vol. 24, pp. 8061-8068, 2020. doi: https://doi.org/10.1007/s00500-019-04153-6.

[19] Gamon M., "Linguistic Correlates of Style: Authorship Classification with Deep Linguistic Analysis Features," *in Proceedings of the 20th International Conference on Computational Linguistics*, Geneva Switzerland, pp. 611-es, 2004. doi:10.3115/1220355.1220443.

[20] Guellil I., Adeel A., Azouaou F., Benali F., and Hachani A., "A Semi-supervised Approach for Sentiment Analysis of Arab(ic+izi) Mes-sages: Application to the Algerian Dialect," *SN Computer Science*, vol. 2, no. 118, pp. 1-18, 2021. doi: 10.1007/s42979-021-00510-1

[21] Ibrahim K., El Habib N., and Satori H, "Sentiment Analysis Approach Based on Combination of Word Embedding Techniques," *in Proceedings of the Embedded Systems and Artificial Intelligence Conference*, Fez, vol. 1076, pp.805-813, 2019. DOI:10.1007/978-981-15-0947-6_76

[22] Jurafsky D. and Martin J., Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Last Visited, 2023. https://web.stanford.edu/~jurafsky/slp3/ed3book.pdf

[23] Kaibi I., Satori H., and Nfaoui E., "A Comparative Evaluation of Word Embeddings Techniques for Twitter Sentiment Analysis," *in Proceedings of the International Conference on Wireless Technologies, Embedded and Intelligent Systems*, Fez, pp. 1-4, 2019. doi: 10.1109/WITS.2019.8723864.

[24] Lample G., Conneau A., Ranzato M., Denoyer L., and Jégou H., "Word Translation Without Parallel Data," *in Proceedings of the 6th International Conference on Learning Representations*, Vancouver, pp. 1-14, 2018. https://doi.org/10.48550/arXiv.1710.04087

[25] Li Y. and Yang T., "Word Embedding for Understanding Natural Language: A Survey," *Guide to Big Data Applications*, vol. 5, no. 2, pp. 48-56, 2013. https://doi.org/10.1007/978-3-319-53817-4_4

[26] Liddy E., SURFACE SURFACE Center for Natural Language Processing School of Information Studies (iSchool) 2001 Natural Language Processing Natural Language Processing Natural Language Processing 1, 2001, https://surface.syr.edu/cnlp, Last Visited, 2023.

[27] Maxwell J., *A Treatise on Electricity and Magnetism*, Oxford: Clarendon, 1892. https://doi.org/10.1017/CBO9780511709333

[28] Mikolov T., Chen K., Corrado G., and Dean J., "Efficient Estimation of Word Representations in Vector Space," *in Proceedings of the 1st International Conference on Learning Representations*, Arizona, pp. 1-12, 2013. https://doi.org/10.48550/arXiv.1301.3781

[29] Mikolov T., Sutskever I., Chen K., Corrado G., and Dean J., "Distributed Representations of Words and Phrases and Their Compositionality," *Advances in Neural Information Processing Systems*, arXiv:1310.4546v1, pp. 1-9, 2013. https://doi.org/10.48550/arXiv.1310.4546

[30] Mikolov T., Yih W., and Geoffrey Z., "Linguistic Regularities in Continuous Space Word Representations," *in Proceedings of the North American Chapter of the Association for Computational Conference of the Linguistics: Human Language Technologies*, Atlanta, pp. 746-751, 2003. doi: 10.3109/10826089109058901

[31] Mohammed A. and Kora R., "Deep Learning Approaches for Arabic Sentiment Analysis," *Social Network Analysis and Mining*, vol. 9, no. 1, pp. 1-12, 2019. doi: 10.1007/s13278-019-0596-4

[32] Mohd M., Jan R., and Shah M., "Text Document Summarization Using Word Embedding," *Expert Systems with Applications*, vol. 143, pp. 112958, 2020. doi: 10.1016/J.ESWA.2019.112958.

[33] Moudjari L., Benamara F., and Akli-Astouati K., "Multi-Level Embeddings for Processing Arabic Social Media Contents," *Computer Speech Language*, vol. 70, pp. 101240, 2021. doi:10.1016/j.csl.2021.101240.

[34] Mulyo M. and Widyantoro D., "Aspect-based Sentiment Analysis Approach with CNN," *in Proceedings of the 5th International Conference on Electrical Engineering*, Malang, pp. 142-147, 2018. doi: 10.1109/EECSI.2018.8752857.

[35] Naaima B., Soumia E., Rdouan F., and Thami R., "Exploring the Use of Word Embedding and Deep Learning in Arabic Sentiment Analysis," *Advances in Intelligent Systems and Computing*, vol. 1105, pp. 149-156, 2020. doi:10.1007/978-3-030-36674-2_16.

[36] Naili M., Chaibi A., and Ghezala H., "Comparative Study of Arabic Stemming Algorithms for Topic Identification," *Procedia Computer Science*, vol. 159, pp. 794-802, 2019, doi: 10.1016/j.procs.2019.09.238.

[37] Nurkasanah A. and Hayaty M., "Feature Extraction Using Lexicon on the Emotion Recognition Dataset of Indonesian Text," *Ultimatics Jurnal Teknik Informatika*, vol. 14, No. 1, pp. 20-27, 2022. doi: 10.31937/TI.V14I1.2540.

[38] Ombabi A., Ouarda W., and Alimi A., "Deep Learning CNN-LSTM Framework for Arabic Sentiment Analysis Using Textual Information Shared in Social Networks," *Social Network Analysis and Mining*, vol. 10, no. 53, pp. 1-13, 2020. https://doi.org/10.1007/s13278-020-00668-1

[39] Pennington J., Socher R., and Manning C., "GloVe: Global Vectors for Word Representation," *in Proceedings of the Empirical Methods in Natural Language Processing Conference*, Doha, pp. 1532-1543, 2014. doi: 10.3115/v1/D14-1162

[40] Ramakrishnan D. and Radhakrishnan K., "Applying Deep Convolutional Neural Network Algorithm in the Cloud Autonomous Vehicles Traffic," *The International Arab Journal of Information Technology*, vol. 19, no. 2, pp. 186-194, 2022. 2021. https://doi.org/10.34028/iajit/19/2/5

[41] Soliman A., Eissa K., and El-Beltagy S., "AraVec: A Set of Arabic Word Embedding Models for Use in Arabic NLP," *Procedia Computer Science*, vol. 117, pp. 256-265, 2017. doi: 10.1016/j.procs.2017.10.117

[42] Statista, "Most common languages used on the internet as of January 2020, by share of internet users," Last Visited, 2023. https://www.statista.com/statistics/262946/share-of-the-most-common-languages-on-the-internet/,

[43] Statista, "The World's Most Spoken Languages," Last Visited, 2023. https://www.statista.com/statistics/266808/the-most-spoken-languages-worldwide/

[44] Suleiman D. and Awajan A., "Comparative Study of Word Embeddings Models and Their Usage in

Arabic Language Applications," *in Proceedings of The 19th International Arab Conference on Information Technology*, Werdanye, pp. 1-7, 2019. doi: 10.1109/ACIT.2018.8672674.

[45] Zyout M. and Hassan N., "Sentiment Analysis of Arabic Tweets about Violence Against Women using Machine Learning," *in Proceedings of the 12th International Conference on Information and Communication Systems Sentiment*, Valencia, pp. 171-176, 2021. doi:10.1109/ICICS52457.2021.9464600.

Ghizlane Bourahouat is Phd Student at The School of Information Sciences, Agdal, Rabat, Morroco. She is a Data Scientist Engineer from the School of Information Science. Her Research Interests Include Artificual Intelligence, Transformers, Natural Language Processing and LLMS.

Manar Abourezq is a Professor at Information Science Avenue Ibnsina B.P. 765 Agdal, Rabat, Morroco. She is an Engineer at the National Institute of Statistics and Applied Economics and has a PhD in Computer Science from ENSIAS. Her Research Interests Include Cloud Computing, Natural Language Processing and Machine Learning.

Najima Daoudi is a full Professor at the school of Information Sciences, Avenue Ibnsina B.P. 765 Agdal, Rabat, Morocco. She is an Engineer at the National School for Computer Science and Systems Analysis and has a PhD in Computer Science from ENSIAS. Her research interests include Ontologies, Artificial Intelligence, NLP, Machine Learning, MLOPS and Data Visualization.