UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Liliane Soares da Costa

OPHELIA - A Neural Solution for Text Classification using joint Embeddings of Words and KG Entities

Florianópolis
2023

Liliane Soares da Costa

OPHELIA - A Neural Solution for Text Classification using joint Embeddings of Words and KG Entities

Tese submetida ao Programa de Pós-Graduação em Ciência da Computação para a obtenção do título de doutora em Ciência da Computação.
Orientador: Prof. Renato Fileto, Dr.

Florianópolis
2023

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Costa, Liliane Soares da
OPHELIA - A Neural Solution for Text Classification using joint Embeddings of Words and KG Entities / Liliane Soares da Costa ; orientador, Renato Fileto, 2023.
93 p.

Tese (doutorado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Ciência da Computação, Florianópolis, 2023.

Inclui referências.

1. Ciência da Computação. 2. Text Classification. 3. Word Embedding. 4. Knowledge Embedding. 5. Deep Neural Networks. I. Fileto, Renato . II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Ciência da Computação. III. Título.

Liliane Soares da Costa

OPHELIA - A Neural Solution for Text Classification using joint Embeddings of Words and KG Entities

O presente trabalho em nível de doutorado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof[a]. Solange Oliveira Rezende, Dr[a].
Universidade de São Paulo

Prof[a]. Valéria Delisandra Feltrim, Dr[a].
Universidade Estadual de Maringá

Prof. Mauro Roisenberg, Dr.
Universidade Federal de Santa Catarina

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de doutora em Ciência da Computação.

Documento assinado digitalmente
PATRICIA DELLA MEA PLENTZ
Data: 28/09/2023 08:18:09-0300
CPF: ***.243.700-**
Verifique as assinaturas em https://v.ufsc.br

Prof[a]. Patricia Della Méa Plentz, Dr[a]
Coordenadora do Programa

Documento assinado digitalmente
Renato Fileto
Data: 28/09/2023 13:25:49-0300
CPF: ***.315.998-**
Verifique as assinaturas em https://v.ufsc.br

Prof. Renato Fileto, Dr.
Orientador

Florianópolis, 2023.

ACKNOWLEDGEMENTS

This study was supported by the Foundation for Research Support of Santa Catarina, Fundação de Amparo à Pesquisa e Inovação do Estado de Santa Catarina (FAPESC), and the Brazilian National Laboratory for Scientific Computing (LNCC), where experiments were conducted using the computational resources of the SDumont supercomputer. I would like to express my sincere gratitude to the many people who provided invaluable assistance during my doctoral studies, including my advisor, Renato Fileto, for his guidance and support throughout these years. Additionally, I am immensely grateful to Italo for his valuable friendship and co-guidance during all stages. I would also like to thank Roque for his support and patience, particularly emotional support and love during this time. I am grateful to my family, especially my mother (may she rest in peace), who encouraged me to pursue a better life through education. Their support, guidance, and encouragement throughout this journey were essential to my success. Finally, I would like to thank God for strengthening me during the challenging times.

RESUMO

A contínua expansão da coleta e disseminação de dados textuais tornou a classificação de texto uma tarefa crucial para aproveitar as enormes quantidades de texto digital disponíveis atualmente. O objetivo da classificação de texto é categorizar um documento de texto em uma ou mais categorias predefinidas dentro de um domínio de aplicação específico. Abordagens existentes de classificação de texto podem ser prejudicadas quando usam apenas o modelo de bag-of-words para representar as características, pois isso ignora a ordem das palavras e os sentidos, que podem variar dependendo do contexto. Os embeddings de palavras surgiram recentemente para superar essas limitações, permitindo melhorias significativas de desempenho ao condensar o conhecimento da linguagem em vetores densos. Além disso, as relações entre entidades do mundo real expressas em grafos de conhecimento podem ser condensadas em vetores densos por meio de embeddings de conhecimento. No entanto, abordagens existentes não aproveitam totalmente os embeddings de conhecimento ao não considerá-las em seus modelos. Modelos tradicionais de representação de texto são limitados, pois focam exclusivamente nas palavras, carecendo da capacidade de diferenciar entre documentos que compartilham o mesmo vocabulário, mas oferecem perspectivas diferentes sobre um determinado assunto. Nesse contexto, este trabalho surge em resposta às diversas aplicações da classificação automática de texto. Além disso, ele se baseia no potencial das representações de espaço vetorial e busca preencher a lacuna relacionada à compreensão da semântica presente em dados de linguagem natural. O principal objetivo deste estudo é avançar a pesquisa no campo da Classificação de Texto, incorporando aspectos semânticos na representação de coleções de documentos. Para isso, propomos OPHELIA, uma abordagem de Rede Neural Profunda (DNN) para tarefas de classificação de texto usando embeddings de conhecimento e palavras. OPHELIA aproveita embeddings conjuntamente treinadas de grafos de conhecimento e texto. Esses embeddings podem fornecer informações contextuais mais consolidadas do que embeddings separados de texto e conhecimento, e seu uso para melhorar a classificação de texto ainda não foi suficientemente explorado. O FastText é usado para treinar embeddings conjuntos de palavras e conhecimento, permitindo que sejam consistentemente integradas em um único espaço incorporado. A rede neural usada para OPHELIA é a Rede Neural Feedforward e a Rede de Cápsulas. Esta tese fornece inicialmente uma revisão abrangente da literatura sobre classificação de texto usando embeddings como características. Em seguida, descrevemos os algoritmos e arquiteturas que compõem OPHELIA. Realizamos experimentos com diferentes modelos de redes neurais profundas com números variados de células e camadas ocultas. Cada arquitetura foi avaliada com sua melhor combinação de parâmetros para comparar seu desempenho com abordagens de ponta. Nossos resultados demonstram que OPHELIA supera as abordagens existentes no conjunto de dados da BBC e permanece competitivo nos conjuntos de dados AG News e Reuters-21578.

Palavras-chave: Classificação de texto. Embedding de palavra. Embedding de conhecimento. Rede neurais profundas.

RESUMO ESTENDIDO

Introdução

A quantidade de dados textuais gerados e armazenados digitalmente tem aumentado em uma taxa exponencial. Esses dados incluem notícias, artigos, postagens em mídias sociais, entre outros. No entanto, esses dados textuais muitas vezes carecem de estrutura e semântica bem definida para fins de processamento computacional, o que pode dificultar o seu uso em diversas tarefas e domínios. Por exemplo, para classificar textos ou recomendar conteúdo personalizado, é necessário que a máquina seja capaz de entender o significado do texto e identificar os tópicos e conceitos relevantes presentes nele. A falta de semântica bem definida e processável por máquina torna essa tarefa mais difícil e pode prejudicar a qualidade dos resultados obtidos. Uma tendência para contornar esses problemas é o uso de técnicas de Inteligência Artificial (IA), especificamente Machine Learning (ML) e Natural Language Processing (NLP), para descobrir automaticamente padrões e classificar documentos de texto. Dada a grande quantidade e diversidade de dados textuais criados diariamente, sua classificação automática, frequentemente em uma variedade de classes possíveis, torna-se crucial para filtrar esses dados para uso em inúmeras aplicações. Para fazer isso com precisão, as relações sintáticas e semânticas entre as palavras, que influenciam seu significado e, consequentemente, o significado geral do texto podem ter que ser levadas em consideração. No entanto, os modelos tradicionais de representação de texto são limitados a palavras, tornando difícil correlacionar textos que usam vocabulário e sintaxe diversos para expressar a mesma ideia ou outras muito semelhantes. Assim, este trabalho é motivado pela complexidade dos conteúdos textuais e pelo uso limitado de recursos semanticamente enriquecidos, particularmente embeddings de palavras e de conhecimento treinados de forma conjunta, para classificação de textos de notícias. Esta tese propõe a abordagem OPHELIA (*knOwledge GraPH-augmented tExt cLassIfication Approach*) baseada em rede neural profunda para classificação de textos utilizando *embeddings* de palavra e de conhecimento. OPHELIA explora *embeddings* de grafos de conhecimento e de texto treinados em conjunto. Estes *embeddings* podem fornecer informações contextuais mais consolidadas do que *embeddings* de conhecimento e de palavras treinados separadamente. O uso desta combinação ainda não foi suficientemente investigado para alavancar abordagens de classificação de textos.

Objetivos

O objetivo geral desta pesquisa é desenvolver e avaliar uma abordagem para classificação de textos, denominada OPHELIA, que utiliza uma rede neural profunda alimentada com *embeddings* de palavras e de conhecimento treinados de forma conjunta. Os objetivos específicos desta pesquisa são: (i) treinar *embeddings* de palavra e conhecimento de forma conjunto através da utilização de textos de notícias e triplas de um grafo de conhecimento; (ii) investigar como diferentes arquiteturas de redes neurais podem explorar os *embeddings* treinados de forma conjunta para classificação de textos; (iii) implementar e avaliar um protótipo de sistema de classificação de textos baseado na abordagem OPHELIA em bases de dados tipicamente utilizadas para avaliar classificadores de textos.

Metodologia

Para validar a hipótese de pesquisa e atingir o objetivo geral, primeiramente foi proposto um processo genérico para a tarefa de classificação de textos. Esse processo foi utilizado como base para o desenvolvimento da arquitetura do OPHELIA. O próximo passo foi treinar os *embeddings* de palavras e de conhecimento de forma conjunta utilizando o fastText. Para isso,

foram utilizados como entrada do fastText triplas de alta qualidade dos infoboxes das entidades para gerar os *embeddings* de conhecimento e os resumos longos das entidades para gerar os *embeddings* de palavras, sendo ambos dados oriundos da DBPedia. Em paralelo, foi realizado o passo de enriquecimento semântico das bases de dados selecionadas para teste, usando-se Babelfy. Para a etapa de classificação de textos, foram propostas duas arquiteturas de redes neurais, uma utilizando a feedforward neural network (FFNN), e outra usando rede neural de cápsula (CapsNet). As redes foram treinadas com os *embeddings* de palavra e de conhecimento treinados de forma conjunta e textos de notícias das bases de dados selecionadas.

Resultados e Discussão

OPHELIA superou algumas abordagens para o conjunto de dados BBC News e apresentou resultados competitivos no conjunto de dados AG News e Reuters-21578. A comparação entre as diferentes estruturas propostas para a rede neural do OPHELIA mostrou que não há diferenças relevantes entre elas. Portanto, é recomendado o uso da FFNN pela facilidade de trenamento e uso de embeddings com menor dimensionalidade. Baseado nos resultados apresentados nesta tese há evidências que o uso de embeddings treinados de forma conjunta trazem resultados competitivos para classificação de textos, se comparados com diferentes abordagens. Há trabalhos da literatura sobre classificação de textos longos e formais que treinam redes neurais profundas com um número significantemente maior de documentos. Enquanto algumas abordagens da literatura utilizam *embeddings* com mais de 300 dimensões, nossos experimentos mostram que a abordagem proposta obtem melhores resultados com *embeddings* de dimensionalidade 50 e 100. Esse é um forte indício que a abordagem OPHELIA pode ser empregada em situações com limitação de *hardware* utilizado no treinamento das redes neurais profundas. O uso de CapsNet adotada na abordagem OPHELIA mostra resultados promissores. Porém, para melhores resultados é interessante realizar experimentos com uma maior variação dos parâmetros da rede.

Considerações Finais

Com base nos resultados obtidos em experimentos, a resposta à pergunta de pesquisa que esta tese busca responder é sim, relações entre embeddings de palavras e de entidades de um grafo de conhecimento fornecem contexto suficiente para classificar textos relacionados a notícias. A abordagem OPHELIA permitiu atingir Acurácia e F1 acima de abordagens do estado da arte no conjunto de dados BBC News, além de resultados competitivos e levemente inferiores nos outros dois conjuntos de dados utilizados nos experimentos.

Palavras-chave: Classificação de texto. Embedding de palavra. Embedding de conhecimento. Rede neurais.

ABSTRACT

The continuous expansion of textual data collection and dissemination has made text classification a crucial task for harnessing the massive amounts of digital text available today. Text classification aims to categorize a text document into one or more predefined categories within a specific application domain. Existing text classification approaches may be hindered when using just the bag-of-words model to represent features because it ignores word order and senses, which can vary depending on context. Word embeddings have recently emerged to address these limitations, allowing for significant performance improvements by condensing language knowledge into dense vectors. Furthermore, real-world entity relationships expressed in knowledge graphs can be condensed into dense vectors through knowledge embeddings. However, existing approaches do not fully leverage knowledge embeddings by failing to consider them in their models. Traditional text representation models are limited as they solely focus on words, lacking the ability to differentiate between documents that share the same vocabulary but offer different perspectives on a given subject. In this context, this work emerges in response to the diverse applications of automatic text classification. Additionally, it builds upon the potential of vector space representations and seeks to bridge the gap related to understanding the semantics present in natural language data. The primary goal of this study is to advance research in the field of Text Classification by incorporating semantic aspects into the representation of document collections. To achieve this, we propose OPHELIA, a Deep Neural Network (DNN) approach for text classification tasks using knowledge and word embeddings. OPHELIA exploits jointly trained embeddings of knowledge graphs and text. These embeddings can provide more consolidated contextual information than separate embeddings of text and knowledge, and their use for enhancing text classification has not been sufficiently explored yet. FastText is used to jointly train word and knowledge embeddings, allowing them to be consistently integrated into a single embedded space. The neural network used for OPHELIA is the Feedforward Neural Network and Capsule Network. This thesis first provides a comprehensive review of the literature on text classification using embeddings as features. Then, we describe the algorithms and architectures that constitute OPHELIA. We conduct experiments with different deep neural network models with varying numbers of hidden cells and hidden layers. Each architecture is evaluated with its optimal parameter combination to compare its performance with state-of-the-art approaches. Our results demonstrate that OPHELIA outperforms existing approaches on the BBC dataset and remains competitive on AG News and Reuters-21578.

Keywords: Text classification. Word Embedding. Knowledge Embedding. Deep neural network.

LIST OF FIGURES

Figure 1 – Three examples of BBC text documents, their respective categories, and some semantic annotations of their contents based on a KG. The grey box represents a concept, and the blue boxes represent named entities. 21
Figure 2 – A reference process for text classification. Adapted from (NAVLANI, 2018). 26
Figure 3 – Example of KG extract (TAILLÉ, 2022). 28
Figure 4 – Example of Semantic annotation. Adapted from (KIRYAKOV et al., 2004). 29
Figure 5 – Example of word embedding representation. 31
Figure 6 – Architecture of a FeedForward Neural Network (MONEDERO et al., 2014) 35
Figure 7 – Architecture of a CapsNet (SABOUR; FROSST; HINTON, 2017) 37
Figure 8 – A capsule network for text classification (KIM et al., 2020) 37
Figure 9 – Steps followed to retrieve the papers analyzed. 39
Figure 10 – Three-dimensional projection of Word2Vec embeddings of some words. . . 43
Figure 11 – Taxonomy of embeddings that can be useful for text classification 43
Figure 12 – Most effective combinations of classification techniques and embeddings identified in the text classification literature. 44
Figure 13 – Overview of OPHELIA main modules and text classification process. 68
Figure 14 – Text classification exploiting joint embeddings of words and knowledge in alternative neural network models: (a) Feed Forward network, the most traditional model for NLP, and (b) the CapsNet-based model (KIM et al., 2020). 71
Figure 15 – Training and valid loss during training of an FFNN. The training lasted 200 epochs. The red line represents the training loss. The green line represents the valid loss. The y-axis is the loss value. The x-axis is the number of epochs. 76
Figure 16 – Training and valid loss during training of a CapsNet. The training lasted 100 epochs. The red line represents the training loss. The green line represents the valid loss. The y-axis is the loss value. The x-axis is the number of epochs. 77
Figure 17 – Confusion matrix obtained on BBC News dataset. 78
Figure 18 – Confusion matrix obtained on AG News dataset. 79
Figure 19 – Confusion matrix obtained on Reuters-21578 dataset. 80

LIST OF SYMBOLS

IDC	International Data Corporation
NLP	Natural Language Processing
EL	Entity Linking
KB	Knowledge Base
KG	Knowledge Graph
ML	Machine Learning
AI	Artificial Intelligence
NER	Named Entity Recognition
DNN	Deep Neural Network
FFNN	Feedforward Neural Network
CapsNet	Capsule Network
OPHELIA	knOwledge GraPH-augmented tExt cLassIfication Approach
RDF	Resource Description Framework
WSD	Word Sense Disambiguation
LP	Link Prediction
LC	Link Classification
ER	Entity Resolutions
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
SVM	Support Vector Machine
GRU	Gated Recurrent Unit
BERT	Bidirectional Encoder Representations from Transformers
DT	Decision Tree
PoS	Part-of-Speech
ROC	Receiver Operating Characteristic
Bi-LSTM	Bidirectional Long Short-Term Memory
XMLC	Extreme multi-label classification
SGD	Stochastic Gradient Descent

CONTENTS

1	**INTRODUCTION**	**19**
1.1	MOTIVATING EXAMPLE	21
1.2	RESEARCH QUESTIONS AND HYPOTHESIS	22
1.3	OBJECTIVES	23
1.4	CONTRIBUTIONS	23
1.5	OUTLINE OF THE MANUSCRIPT	24
2	**FOUNDATIONS**	**25**
2.1	TEXT CLASSIFICATION	25
2.2	KNOWLEDGE GRAPH	27
2.3	SEMANTIC ANNOTATION TASKS	28
2.4	EMBEDDINGS	30
2.4.1	**Word Embedding**	**31**
2.4.2	**Knowledge Embedding**	**32**
2.4.3	**Joint Embeddings of Words and Knowledge**	**33**
2.5	NEURAL NETWORKS FOR TEXT CLASSIFICATION	35
2.5.1	**Feedforward Neural Network**	**35**
2.5.2	**Capsule Neural Networks**	**36**
3	**TEXT CLASSIFICATION USING EMBEDDINGS: SURVEY AND DIRECTIONS**	**39**
3.1	BIBLIOGRAPHICAL REVIEW PROCEDURE	39
3.2	SEMANTICS AND EMBEDDINGS IN TEXT CLASSIFICATION	40
3.2.1	**Key aspects of text classification approaches**	**40**
3.2.2	**Embeddings as features for text classification**	**41**
3.2.3	**Effective Combinations of Classification Techniques with Embeddings**	**44**
3.3	CURRENT TEXT CLASSIFICATION APPROACHES USING EMBEDDINGS	45
3.3.1	**Approaches using Word2Vec**	**45**
3.3.2	**Approaches using GloVe**	**48**
3.3.3	**Approaches using BERT**	**50**
3.3.4	**Approaches using Other Embeddings**	**51**
3.3.5	**Approaches using Embedding Combinations**	**53**
3.4	COMPARATIVE ANALYSIS OF TEXT CLASSIFICATION APPROACHES	56
3.5	EVALUATION OF TEXT CLASSIFICATION APPROACHES	58
3.5.1	**Directions for Text Classification Research**	**58**
4	**THE OPHELIA APPROACH**	**67**

4.1 EMBEDDING GENERATION MODULE 67
4.2 SEMANTIC ENRICHMENT MODULE 69
4.3 TEXT CLASSIFICATION MODULE 69

5 EXPERIMENTAL EVALUATION 73
5.1 EXPERIMENTAL SETUP 73
5.2 NEURAL NETWORK EVALUATION 76
5.3 OPHELIA EVALUATION 76
5.4 DISCUSSION 79

6 CONCLUSION 83

BIBLIOGRAPHY 85

1 INTRODUCTION

Digital data is growing at an exponential rate, and its use is transforming the way people live and work. This increasing volume of digital data is primarily attributed to the proliferation of the internet and mobile devices. The International Data Corporation (IDC)[1] predicts that by 2025 worldwide data creation will grow to 175 zettabytes (RYDNING; REINSEL; GANTZ, 2018). The internet has enabled the creation of vast amounts of data from social media, online transactions, and digital communication. The ability to collect, store, and analyze this data has opened new avenues for innovation, allowing businesses to personalize their products and services and governments to improve public services. However, the complexity goes beyond the fast increase in data volumes. Data originates from multiple sources, exists in different models and formats, and is stored in various locations.

Textual data is a significant part of the expanding digital data amount (KUMAR; KAR; ILAVARASAN, 2021). However, despite the abundance of this data, its effective utilization by applications is hindered by the challenge of capturing the precise semantics of its content and identifying relevant documents or text passages that satisfy specific topics or needs. To overcome these hurdles, there is a growing trend to leverage Artificial Intelligence (AI) techniques, such as Machine Learning (ML) and Natural Language Processing (NLP), to uncover patterns and classify text documents automatically.

Given the vast and diverse textual data generated daily, automatic classification into categories is crucial for screening these data in numerous applications (DENG et al., 2019). To achieve this, the syntactic and semantic relations between words and named entities mentioned in a text must be considered. However, traditional models for text representation are limited to words, making it challenging to differentiate documents that express distinct perspectives on a subject but use similar vocabulary and syntax. Conversely, it is also difficult to correlate texts that utilize different vocabulary and syntax to express very similar ideas. Thus, this thesis addresses the complexity of textual contents and the limited use of semantically rich features, particularly joint embeddings of words and knowledge, for text classification.

Currently, several text classification approaches employ embeddings as features to classify texts with success (LENC; KRÁL, 2017; SINOARA et al., 2019; ZHANG; LERTVITTAYAKUMJORN; GUO, 2019; PITTARAS et al., 2021; LEE; LEE; YU, 2021; ZHANG; YAMANA, 2021). Most of these classification approaches are based on Deep Neural Networks (DNNs). Word embeddings are gaining popularity because they encode semantic and syntactic properties of words into compacted vectors based on the local context in which they typically appear in the texts used to train the embedding model. Consequently, embeddings help improve the accuracy of text classification while maintaining scalability for processing vast amounts of text data.

In addition to word embeddings, text classification can benefit from semantics derived

[1] https://www.idc.com/

from embeddings trained on Knowledge Graphs (KGs) such as DBpedia[2] (AUER et al., 2007; LEHMANN et al., 2009) and Yago[3] (FABIAN; GJERGJI; GERHARD, 2007). According to (NICKEL et al., 2016a), KGs are knowledge representations of Knowledge Bases (KBs) that capture information as entities and semantic relations between them. The relations between entities in a KG captured by such embeddings, called knowledge embeddings, may provide additional knowledge about an entity mentioned in a text, thus enriching its representation for text classification.

Word embedding and knowledge embedding techniques aim to represent, respectively, words and entities in n-dimensional continuous vector space. Word embeddings (LI; YANG, 2018) trained with large volumes of text capture relations between words. Knowledge embeddings (WANG et al., 2017), on the other hand, capture relationships, which can be represented as triples in some KG, between unambiguous entities. The additional knowledge from knowledge embeddings could benefit text classification. DNNs have been successfully used with embeddings for text classification, among other tasks, because they can capture linear and non-linear relations between embeddings. Thus, we also can exploit knowledge embeddings as well as text embeddings. However, we have not found any works that exploit the combined semantics of words and the semantics of KG entities for text classification. Among the few text classification approaches that employ knowledge embeddings and DNNs (SINOARA et al., 2019; ZHANG; LERTVITTAYAKUMJORN; GUO, 2019), none use these embeddings in a combined way for text classification. Moreover, several works that use word embeddings as features for text classification employ vectors with 300 or more dimensions (WANG et al., 2016; GARGIULO et al., 2019; SINOARA et al., 2019). The higher the number of dimensions of embedding representations, the higher the hardware requirements to train and run approaches based on embeddings. Thus, besides the potential benefits in classification capabilities and results quality, the combination of word and knowledge embeddings has the potential to decrease the number of dimensions currently being employed and, consequently, decrease the hardware requirements and improve the approach scalability. It may enable text classification approaches based on these combined embeddings to be applied under more strict computational and financial limitations.

This thesis proposes OPHELIA - knOwledge GraPH-augmented tExt cLassIfication Approach. OPHELIA is a neural network-based approach that exploits embeddings of words and knowledge in a shared space to tackle the text classification task. Firstly, it jointly trains word embeddings and knowledge embeddings using fastText (JOULIN et al., 2016; JOULIN et al., 2017b). Then, OPHELIA employs these embeddings to represent ordinary words and entities for each recognized mention in the text documents. Different from other approaches, OPHELIA uses an entity recognition and linking tool to semantically annotate named entity mentions found in the text and replaces these mentions with their respective entities in a KG. Thus, it uses embedded representations of words (word embeddings) and entities (knowledge

[2] https://wiki.dbpedia.org
[3] http://www.yago-knowledge.org/

embeddings) as features fed to neural network classifiers. Experiments with text classification benchmarks based on public datasets of news articles show the viability and benefits of our approach, which achieves accuracy and F1 measures comparable to those of state-of-the-art approaches on the AG News and Reuters dataset and outperforms them on the BBC News dataset.

1.1 MOTIVATING EXAMPLE

Semantics is crucial for text classification in certain circumstances. Semantic annotation tasks such as entity linking (EL) (SHEN; WANG; HAN, 2015a; OLIVEIRA et al., 2021) and word sense disambiguation (WSD) (NAVIGLI, 2009a) could be applied to extract such features. Figure 1 shows three text documents taken from the BBC News Website[4], with some of their words (in bold) disambiguated to specific concepts or named entities (represented by rectangles), which can be described in a knowledge graph (KG) (ALTINEL; GANIZ, 2018; ALY; REMUS; BIEMANN, 2019). These facts are represented as triples of the form $\langle node, link, node \rangle$, in which each node refers to a concept (class), named entity (instance), or literal, and each link represents a particular semantic relationship between nodes. The semantic annotations (linking words to KG nodes) in Figure 1 were created by submitting the texts to the annotation tool Babelfy[5], which linked words of the texts to nodes of the KG Babelnet[6].

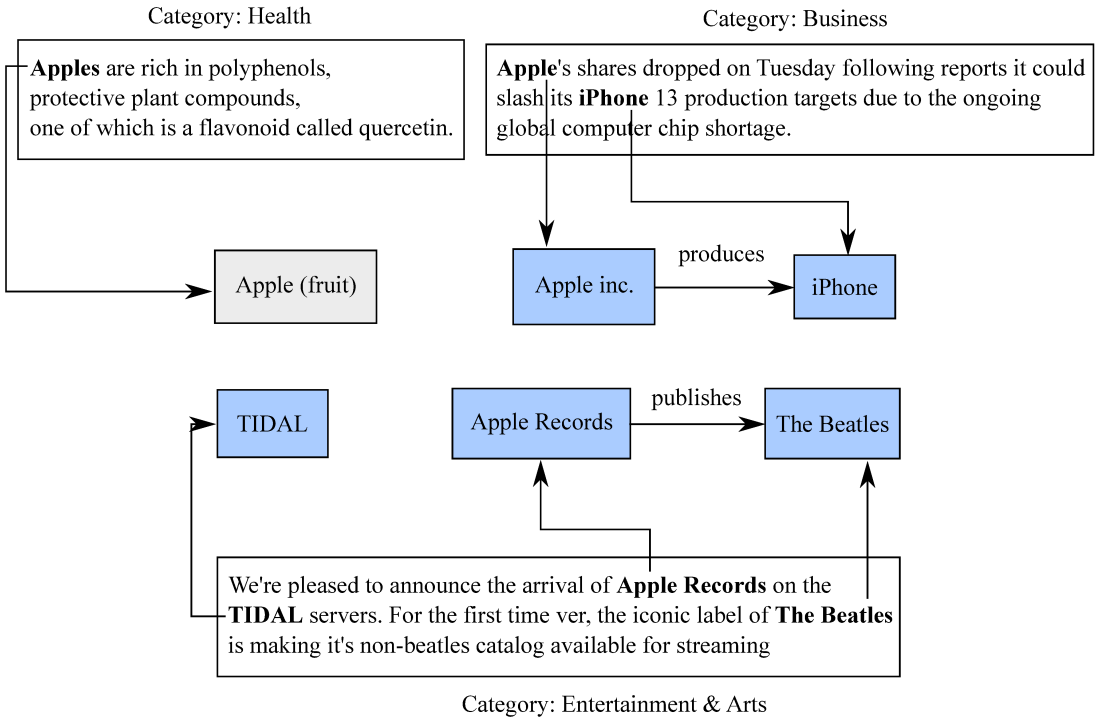

Figure 1 – Three examples of BBC text documents, their respective categories, and some semantic annotations of their contents based on a KG. The grey box represents a concept, and the blue boxes represent named entities.

[4] https://www.bbc.com/news
[5] http://babelfy.org/
[6] https://babelnet.org/

The word stem “Apple” appears in the three documents but refers to a distinct thing in each one. Its disambiguation can be done according to its respective text context. In the text at the top left corner of Figure 1, the word “Apples” refers to the *fruit*, which is said to be rich in some chemicals that can be found in certain kinds of food. Therefore, it is disambiguated to the word represented by the grey box, which is described in WordNet[7]. Considering this link and the remaining text, it is possible to classify this text in the category *Health*. On the other hand, the word “Apple” in the text at the top right was linked to the entity *Apple inc.*, which is a technology company, as the text also mentions the product *iPhone*, produced by *Apple inc.*, as represented by the link between them. It is possible to use these semantic annotations to classify this text in the category *Business*. Lastly, the text at the bottom was classified as *Entertainment & Arts* because several of its words were linked to entities related to music, namely the streaming service *TIDAL*, the record label *Apple Records* and the rock band *The Beatles*. The categories used to classify these texts are those of the BBC News Website.

This example illustrates how precise semantics can be essential for text classification in challenging situations, like classifying short texts in a wide range of categories. Correctly capturing subtle semantic distinctions from texts with little contextual information can be essential for proper classification.

1.2 RESEARCH QUESTIONS AND HYPOTHESIS

As the existing approaches for text classification do not fully exploit the context provided by entity and word relations, this thesis aims to answer the following research question:

Does the exploitation of relations between words and entities of a KG provide a context that allows for better text classification results than existing approaches?

To answer this research question, we envision that jointly trained embeddings of words and knowledge can offer a more semantic context and enable better results for news text classification than existing approaches, which employ just one or none of these embeddings. The joint training of the embeddings allows us to take advantage of relations between words and entities. A neural network may effectively exploit linear and non-linear relations between word and knowledge in a joint embedded space, making further reward from a more holistic semantic context. Therefore, the hypothesis of this work is: *a neural network-based text classification approach that exploits jointly trained embeddings of word and knowledge can achieve a higher accuracy and F1 score for the text classification task than the existing approaches.* We use the accuracy and F1 score because it is the most commonly used metric to evaluate text classification approaches, as shown in Section 3.5.

[7] http://wordnetweb.princeton.edu/perl/webwn?s=apple

1.3 OBJECTIVES

The general objective of this research is to develop a text classification system called OPHELIA (knOwledge GraPH-augmented tExt cLassIfication Approach) that utilizes jointly trained embeddings of words and knowledge to classify text documents using a neural network. The specific goals of this research are:

1. Jointly train word and knowledge embeddings using triples from a KG;
2. Investigate how neural networks can exploit jointly trained word and knowledge embeddings to classify texts;
3. Propose and develop an approach for text classification that incorporates semantics into the representation of textual data and leverages semantic information of different natures.

1.4 CONTRIBUTIONS

OPHELIA employs word and knowledge embeddings jointly trained using the fastText technique (detailed in Section 2.4.3) in two different neural network architectures, allowing OPHELIA to tackle text classification in different datasets. OPHELIA stays competitive with other approaches in the dataset in which the neural network was trained. Based on this thesis proposal and the results, the contributions of this thesis are:

1. A comprehensive survey of text classification approaches that use embeddings as features;
2. A novel approach based on neural networks that exploits jointly trained word and knowledge embeddings in the text classification task.
3. A text classification system called OPHELIA, based on the proposed approach, that achieves competitive results on public datasets;

The contributions mentioned above have resulted in two publications in a scientific journal and a conference specialized in the area of computational intelligence. These publications are:

COSTA, Liliane Soares; OLIVEIRA, Italo Lopes; FILETO, Renato. A Neural Network Approach for Text Classification Using Low Dimensional Joint Embeddings of Words and Knowledge. In: International Conference on Information Integration and Web. Cham: Springer Nature Switzerland, 2022. p. 181-194. DOI: https://doi.org/10.1007/978-3-031-21047-1_17

COSTA, Liliane Soares; OLIVEIRA, Italo L.; FILETO, Renato. Text classification using embeddings: a survey. Knowledge and Information Systems, v. 65, n. 7, p. 2761-2803, 2023. DOI: https://doi.org/10.1007/s10115-023-01856-z

1.5 OUTLINE OF THE MANUSCRIPT

The remaining of this thesis is structured as follows. Chapter 2 presents the basic concepts necessary to understand this thesis. Chapter 3 reviews literature about text classification approaches related to this thesis. Chapter 4 details our OPHELIA approach for text classification. Chapter 5 reports the experiments carried out to evaluate our approach and discusses their results. Finally, Chapter 6 presents the conclusions of this thesis.

2 FOUNDATIONS

This chapter presents the basic concepts necessary to understand the rest of this work. First, Section 2.1 presents the text classification process. Then, Section 2.2 and Section 2.3 present, respectively, the basic concepts of knowledge graphs and the semantic annotation task. Section 2.4 presents the concept of embeddings, including knowledge and word embeddings. Finally, Section 2.5 presents the neural network architectures employed in this text classification approach.

2.1 TEXT CLASSIFICATION

Text classification (also known as text categorization) involves assigning tags or categories to text documents based on their contents (SEBASTIANI, 2002). Formally, given a set of documents D and a set of predefined categories C, the problem of text classification can be modeled as finding a mapping function F from the Cartesian product $D \times C$ to a set $True, False$, i.e., $F : D \times C \rightarrow True, False$. This mapping function F is called a classifier. For example, based on this mapping, for a document $d_i \in D$ and a category $c_j \in C$, if $F(d_i,\ c_j) = True$, then d_i belongs to category c_j, otherwise d_i does not belong to c_j (DENG et al., 2019)[1].

Text classification has various applications, including sentiment analysis, topic labeling, spam detection, and intent detection (ALTINEL; GANIZ, 2018). It can be used to classify short texts, such as tweets and headlines, as well as much larger documents like books, articles, news, and legal contracts. There are two main approaches to text classification: manual and automatic. In manual classification, a human annotator reads and interprets the text content to assign it to a category. While this approach yields accurate results, it can be time-consuming and costly. Automatic classification, on the other hand, uses NLP, machine learning, and other techniques to categorize text automatically, making it a faster and more cost-effective method.

The text classification task has been approached in various ways, utilizing different algorithms and techniques, as evidenced by numerous studies and research papers. However, supervised learning has gained popularity as a means to accomplish this task. This approach involves using a set of pre-classified documents to train a classification model that can subsequently be applied to classify new, unseen documents (DIAB; HAMAYDEH, 2019). The most common statistical and machine learning techniques used for text classification in the literature include the kNN method (CUNNINGHAM; DELANY, 2007), Naive Bayes (DOMINGOS; PAZZANI, 1997), multivariate regression models (YANG; CHUTE, 1994), decision trees (QUINLAN, 1986), Support Vector Machines (SVMs) (VAPNIK, 1999), neural networks (JOHNSON; ZHANG, 2014), the graph partitioning-based approach (GAO et al., 2005), and genetic algorithm-based methods (PIETRAMALA et al., 2008).

[1] Notice that this definition allows for the classification to be multi-class ($|C| > 2$) and multi-label (one document belonging to more than one label), although it can be restricted to binary and single-label classification.

The process of text classification is crucially important in ensuring the accuracy of the outcome. It involves two fundamental stages: training and usage. During the training stage, the system builds a knowledge base using a set of examples. In the next stage, the usage phase, the system uses the knowledge acquired during the training phase to categorize documents with unknown classifications. Figure 2 provides an overview of a standard reference process for text classification.

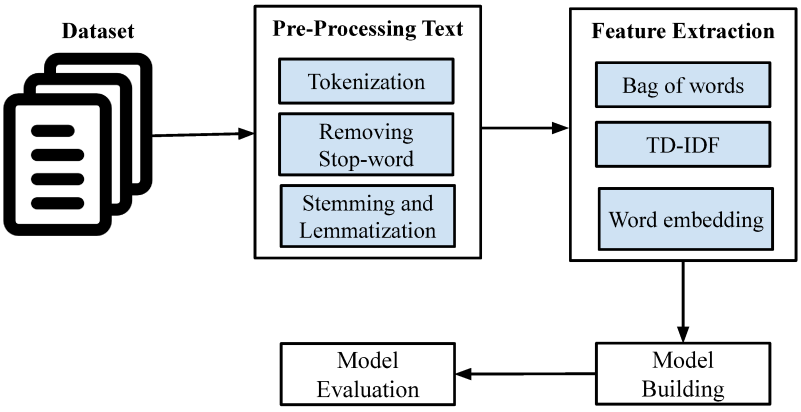

Figure 2 – A reference process for text classification. Adapted from (NAVLANI, 2018).

Text classification typically takes a dataset of text documents as input, which may be in various formats and may not necessarily be previously classified. The documents are first pre-processed, which includes steps such as tokenization, cleaning, and standardization. The next step is feature extraction, where meaningful features are extracted from the text. These features and the associated categories of the documents are then used to learn a classification model, which is evaluated for performance before being used to classify unknown data (KORDE; MAHENDER, 2012).

During the pre-processing step, the text is typically tokenized, which involves breaking down the text into individual words or phrases. The text may also be cleaned by removing stop-words, which are common words that do not carry much meaning, and by removing punctuation and other non-alphabetic characters. The text may be standardized by converting all characters to lowercase or removing numbers. Further normalization is achieved through techniques such as stemming and lemmatization. Stemming algorithms convert different word forms to their root form, while lemmatizers use a language's complete vocabulary for Part-of-Speech (PoS) tagging and provide more informative lemmas. PoS tagging can also be used to identify stop-words for removal. These techniques help to reduce the number of unique words in the text, making it easier to extract meaningful features.

The feature extraction step of text classification uses techniques like bag-of-words ((WALLACH, 2006)), TF-IDF ((HAVRLANT; KREINOVICH, 2017)), and word embeddings ((KÖHN, 2015)) to extract relevant information from documents and create vector representations of their contents. These vectors are used as inputs for machine learning methods to build classification models. Therefore, feature selection aims to identify and select the most

important attributes from a collection of documents by using a predetermined measure of word importance. Good features are essential for model performance, but some representation methods have limitations like high dimensionality and loss of semantic relationships between words.

After selecting relevant features, a classification model can be trained using various machine learning methods, including unsupervised, supervised, or semi-supervised techniques (KOWSARI et al., 2019). Text classification has been extensively studied, and machine learning algorithms have shown promising results. However, there are still challenging issues, such as multi-class classification, unbalanced class distributions, and limited labeled data for training the model.

Finally, the experimental evaluation of classifiers is usually performed by measuring the classification performance (SEBASTIANI, 2002; LI et al., 2022). Several performance measures can be calculated based on the values of a confusion matrix, which presents the number of instances correctly and incorrectly classified by a given classifier. Common performance measures used to evaluate classifiers include accuracy (measures the proportion of correctly classified instances out of all instances); precision (measures the proportion of true positives out of all predicted positive instances); recall (measures the proportion of true positives out of all actual positive instances); F1-score (measure that combines both precision and recall into a single value), and area under the receiver operating characteristic (ROC) (measure that plots the true positive rate against the false positive rate at various classification thresholds, and calculates the area under the resulting curve).

2.2 KNOWLEDGE GRAPH

A Knowledge Graph (KG) is a collection of interconnected facts, represented as a directed graph, where nodes represent entities, and edges represent relationships between entities (NICKEL et al., 2016a; WANG et al., 2017). The relationships between entities are expressed in the RDF format as binary relationships of the form (*subject*, *predicate*, *object*), where the *subject* refers to an entity, the *object* refers to an entity or a literal (e.g., string, number), and the *predicate* represents the relationship between them. Therefore, a KG can be seen as a collection of facts that are structured as a graph. Definition 2.2.1 presents a more formal definition proposed by Wang et al. (2017).

Definition 2.2.1 (Knowledge Graph) *Given a set of entities E and a set of relations R, a Knowledge Graph KG is a set of observed facts F^+. Each fact is represented as (e_i, r_k, e_j), where $e_i, e_j \in E$ are, respectively, the subject and the object, and $r_k \in R$ is the predicate. The set of non-observable facts, i.e., facts that are false or unknown, is represented as F^-. The union of the observed and non-observable facts ($F^+ \bigcup F^-$) is represented as F.*

Figure 3 shows a small portion of a KG containing facts about *Steve Jobs*. Facts are thus stored as triples such as (Apple Inc., founded by Steve Jobs). Entity and Relations types

are predefined, following a format of knowledge representation. It is worth noting that although some of the objects in these facts are literals (such as the relations *gender* and *occupation*), Definition 2.2.1 considers these literals as entities for the sake of simplicity.

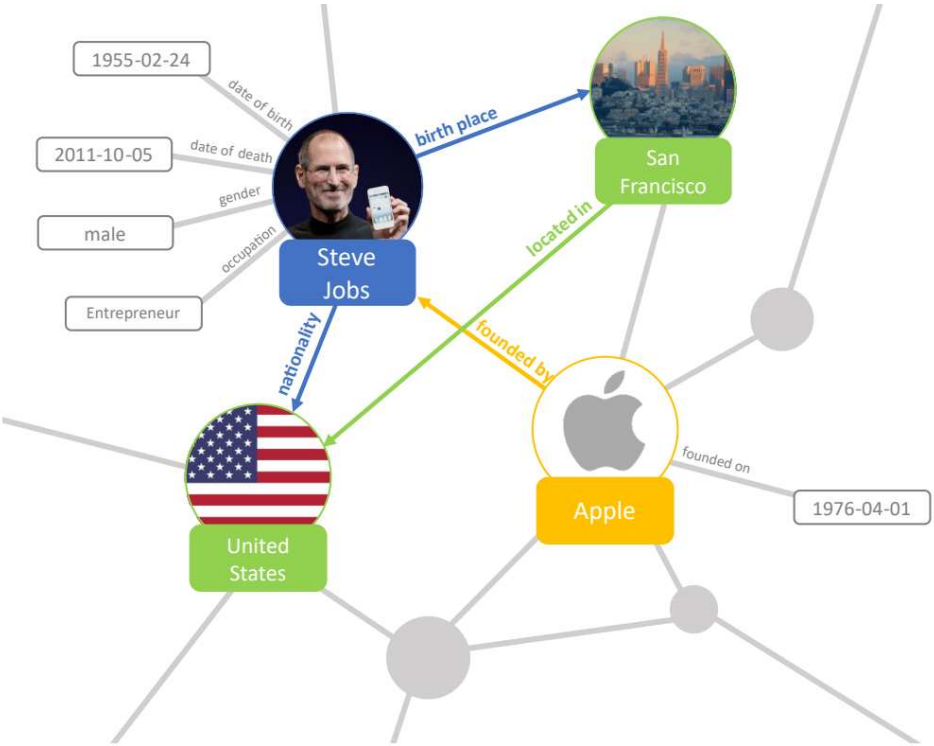

Figure 3 – Example of KG extract (TAILLÉ, 2022).

We have chosen to formalize a KG as a collection of facts expressed as triples instead of as a graph for two reasons: (i) a collection of facts can express a graph without loss of information, and (ii) this formalization can seamlessly be integrated with the Knowledge Embedding discussion (Section 2.4).

A KG is typically built on top of existing databases to link their data together at web-scale. It can combine both structured and unstructured data from several data sources. Connecting datasets in a meaningful way is strategic for business as it helps decision-makers, users, and computers to gain context within the existing knowledge of an organization. The relationships between entities can help to understand the context of a query and to assign specific meaning to user intents.

2.3 SEMANTIC ANNOTATION TASKS

Semantic annotations aim to semantically enrich data to enable new applications of that data (e.g., new access methods, semantic-enabled data analysis, and classification) and to extend existing ones (KIRYAKOV et al., 2004). Formally, the semantic annotation of text documents identifies entity mentions in their contents and links these mentions to precise descriptions of the entities they refer to. This formal process enables machines to better understand the context

and meaning of the text, ultimately improving the accuracy and efficacy of various applications that utilize the annotated data.

The process involves assigning entity mentions in the text to links of their semantic descriptors that can be available in a Knowledge Base (KB) in the form of a KG, as illustrated by Figure 3. In this figure, semantic annotations are represented by dotted curved arrows from entity mentions identified by distinct colors in the text at the top of the figure to entity descriptors in the KG snippet presented at the bottom of the figure. Semantic annotation can eliminate possible ambiguities of the textual mention by connecting it to a precise entity descriptor, which can, in turn, be linked via known facts present in a KB to several other entities. For instance, linking the mention "XYZ" to its descriptor in the KG of Figure 3 eliminates possible ambiguities about which "XYZ" it refers to. The facts about this entity in the KG elucidate that it is a "Company" headquartered in "London", the "City" that is part of the "UK" (not any other city or place with this name), and so on.

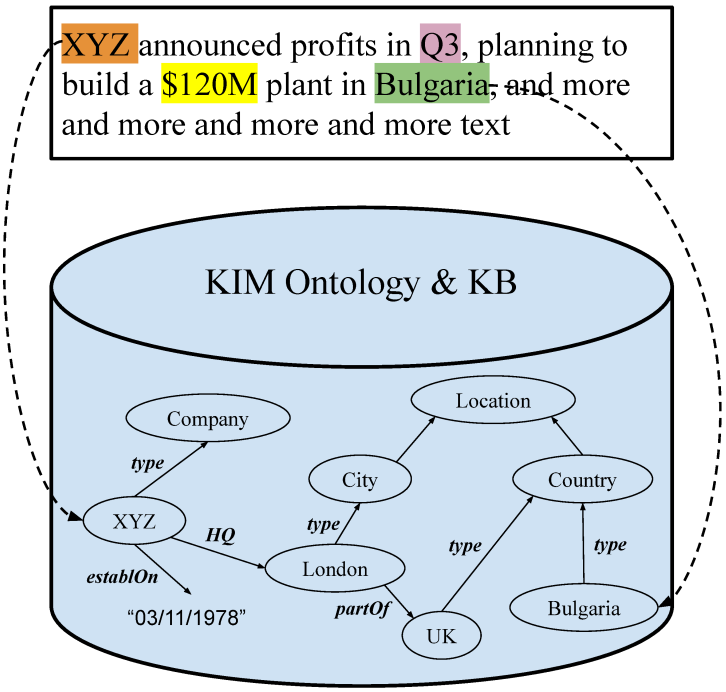

Figure 4 – Example of Semantic annotation. Adapted from (KIRYAKOV et al., 2004).

The process of semantic annotation involves the utilization and creation of metadata and can be seen as a cyclical undertaking that involves both the act of annotating documents and the creation and modification of a domain ontology. The incoming information that requires annotation necessitates not only further annotation but also an ongoing adjustment to new semantic terms and connections. This process facilitates various applications, such as highlighting, indexing and retrieval, categorization, and the generation of more advanced metadata and pertinent knowledge. Additionally, the extraction of more intricate dependencies -

including analysis of relationships between entities, event and situation descriptions, and so on - can facilitate knowledge acquisition.

There are several semantic annotation tasks. This thesis focuses mainly in the EL and the WSD tasks. Thus, we provide formal definitions for both tasks as follows. Definition 2.3.1 is taken from (NADEAU; SEKINE, 2007; SHEN; WANG; HAN, 2015b), while Definition 2.3.2 is taken from (NAVIGLI, 2009b).

Definition 2.3.1 (Entity Linking) *Given a KG **K**with a set of semantic entities **V** and a text document **T**, the EL task aims to recognize a set entity mentions $\boldsymbol{M} \in \boldsymbol{T}$ and link each entity mention $m \in \boldsymbol{M}$ to the semantic entity $v \in \boldsymbol{V}$ that best describes **m**, according with the context where **m** appears in **T**.*

Definition 2.3.2 (Word Sense Disambiguation) *Given a lexicon **L** with a set of synsets (a group of data elements that are considered semantically equivalent) **S** and a text document **T** containing a set of relevant words $\boldsymbol{W} \in \boldsymbol{T}$, the WSD task aims to link each relevant word $w \in \boldsymbol{W}$ to its correct synset $s \in \boldsymbol{S}$ that best describes w, according with the context where w appears in **T**.*

These definitions reinforce that EL and WSD tasks are similar. Both tasks try to establish links between text portions and descriptors of things that they refer to. Such a descriptor can be a node in a KG, for example, which can be represented as a tensor in an embedded space for convenience and fast processing.

2.4 EMBEDDINGS

Embeddings involve projecting data features into a lower-dimensional space. They are utilized for encoding complex information, including graphs and words, into condensed formats ideal for efficient processing, such as low-dimensional continuous vector spaces(GOYAL; FERRARA, 2018). Ideally, an embedding should represent the data without losing information because they are designed to maintain the fundamental characteristics of the original data. A high-quality embedding would result in the exact same data upon conversion back to its original representation (NICKEL et al., 2016a). Nevertheless, real-world scenarios may introduce losses and distortions to the original data. Therefore, the focus is on retaining the significant features (e.g., structural and/or semantic aspects) in the embedded representation for a particular purpose.

An embedding should capture the semantics of the data by placing semantically similar things close to each other in the embedding space. For example, books from the same author and literary genre (e.g., science fiction, drama) should be close to each other in the embedding space. This characteristic of embeddings is useful for several tasks, such as classification (e.g., text, nodes of a graph (GOYAL; FERRARA, 2018)), matching (e.g., Entity Linking (EL) (ZHU;

IGLESIAS, 2018), Entity Resolution (ER) (NICKEL; TRESP; KRIEGEL, 2011)). Besides these tasks, embeddings have mainly been employed in approaches based on machine learning.

Embedding functions are the standard and effective way to transform such discrete input objects into useful vectors in a continuous compact space (GOYAL; FERRARA, 2018). Machine learning techniques such as neural networks, which work on vectors of real numbers, are currently some of the best means to produce embeddings. They can be trained so that all values of a set of features contribute to produce dense vectors with desirable properties. However, many important inputs, such as words of text, do not have a natural vector representation.

Since embeddings map objects to vectors, applications can use similarity in vector space (for instance, Euclidean distance or the angle between vectors) as a robust and flexible measure of object similarity. One common use is to find the nearest neighbors. Although embeddings can represent a myriad of data types (e.g., text, images, sound, graphs), in this work, we focus on the embedding representation of two types of data: words and entities of a KG.

2.4.1 Word Embedding

Word embeddings are a way to represent individual words as vectors in a high-dimensional space. Formally, given a set of words W, a word embedding can be defined as a function $W : words \rightarrow \mathbb{R}^n$ that maps words to n-dimensional real-valued vectors, typically ranging from 50 to 500 dimensions. These vectors encode both semantic and syntactic information about words in a format that computers can process. The information is derived from the contexts in which the words appear in a corpus, which is typically a large collection of text used to train the embedding. Each word in the corpus is mapped to a unique vector in the embedding space, as shown in an example in Figure 5.

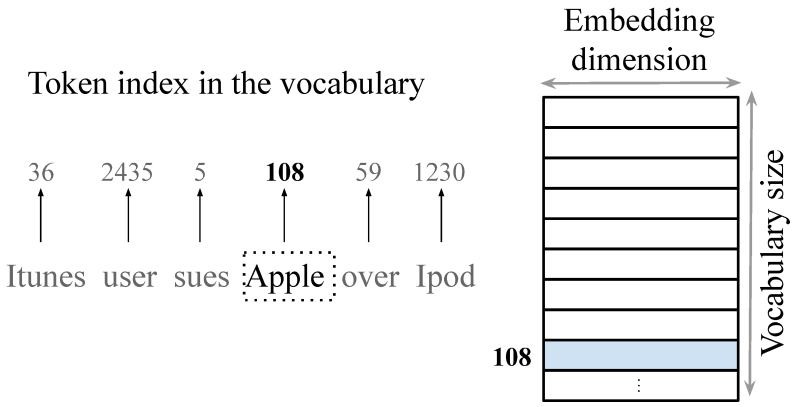

Figure 5 – Example of word embedding representation.

The embedding technique is often associated with the field of deep learning. This approach was first introduced by Bengio et al. (2003) but gained extreme popularity with word2vec (MIKOLOV et al., 2013a). There are also other variants of word embeddings, such

as GloVe (PENNINGTON; SOCHER; MANNING, 2014) and fastText (BOJANOWSKI et al., 2017). These approaches have been the most prominent models of word embeddings. All of them are unsupervised and take a corpus or dataset as input to generate word vectors as output.

Among several methods for creating word embeddings, one of the most common techniques is to learn the meaning of words based on the words that appear nearby in text. This is because words that are used in similar ways tend to have similar meanings, and by capturing this relationship between words, we can generate embeddings that naturally capture their meaning.

This is in contrast to a bag-of-words model, where each word is represented as a separate feature, and different words have different representations, even if they are used in similar ways. This representation is often considered "crisp but fragile" because it does not capture the relationship between words in a way that reflects their actual meaning. By contrast, the distributed representation used in word embeddings captures the meaning of words in a more nuanced and effective way.

One advantage of word embeddings is that they preserve the similarity between words, even in low-dimensional spaces. Similar words have similar vectors, making it possible to perform operations such as computing word similarity and even analogies. These operations are efficient and fast due to the compactness of the vector representation. In summary, word embeddings are a powerful tool for representing words in a way that captures their semantic and syntactic properties, making it possible for computers to process language more naturally and effectively.

2.4.2 Knowledge Embedding

Knowledge embeddings encode KG vertices and edges in continuous vector spaces, preferably low-dimensional ones, with minimal loss of structure and semantics. Despite the advantages of word embeddings, they are usually unsuitable for coping with ambiguity and do not take advantage of the semantic relationships between word meanings and entities in KGs. Thus, KG embeddings can be a better choice for representing relevant features for text classification in some situations. Analogous to word embeddings, KG embeddings aim to represent entities and relations from a KG in continuous vector spaces. This allows the representation of a fact $f_l \in F$ in such way that simplifies the manipulation of a KG without losing useful semantic and structural properties. Most of the KG embedding techniques focus on observed facts, i.e., facts that exist in a KG ($f_l \in F^+$).

According to Wang et al. (2017), KG embedding techniques typically involve three steps: (i) representing entities and relations, (ii) defining a score function, and (iii) learning embedded representations of entities and relations. Step (i) refers to how entities and relations are represented in continuous vector spaces. For several techniques (NICKEL; TRESP; KRIEGEL, 2011; BORDES et al., 2013; WANG et al., 2014; LIN et al., 2015), entities are represented as vectors, while relations are represented as operations in vector spaces. Such representation refers to the distinct facts $f_l \in F^+$ that entities and relations take part in a KG. The scoring

function in (ii) measures the plausibility of a fact f_l, i.e., the probability of the fact f_l to be true. Existing facts in a KG present a higher plausibility than facts that do not exist or do not have been discovered yet. Lastly, (iii) is an optimization problem that maximizes the plausibility of the existing facts in a KG to learn embedded representations of entities and relations. According to Nickel et al. (2016b), the learning representation of entities and relations proposed by KG embedding techniques is a *supervised representation learning* problem.

KG embedding techniques efficiently address KG-related tasks such as link prediction (LP), link classification (LC), and entity resolution (ER), among others (WANG et al., 2017). However, this work focuses only on the embeddings themselves and, therefore, does not discuss KG-related tasks.

2.4.3 Joint Embeddings of Words and Knowledge

Knowledge embeddings and word embeddings are two types of vector representations used in natural language processing tasks, as it was explained in previous subsections (2.4.1 and 2.4.2). Word embeddings are commonly used to represent words as vectors in a high-dimensional space, where each dimension represents a different semantic or syntactic feature. Knowledge embeddings, on the other hand, are representations of concepts or entities in a structured knowledge base, such as a knowledge graph.

By combining these two types of embeddings, we can improve the representation of both. This is because knowledge embeddings capture information about relationships and hierarchies between concepts, while word embeddings capture information about the context in which words are used. This allows, for example, the representation of the named entity “Paris” to be close to that of the word “France”. However, to ensure that their representations have a similar meaning, the knowledge embeddings and the word embeddings must be in the same vector space. This can be done by aligning the embeddings (WANG et al., 2014; ZHONG et al., 2015) or by training them jointly, as done by fastText (BOJANOWSKI et al., 2017; JOULIN et al., 2017a; JOULIN et al., 2017b), which is used in our proposal.

The alignment of knowledge and word embeddings is a process used in NLP to combine structured data from knowledge graphs with unstructured text data, with the aim of improving the performance of NLP applications. This is achieved by using a knowledge model to generate the knowledge embeddings, a word model to generate the word embeddings, and an alignment model that maps the two types of embeddings into the same vector space. The alignment model uses some features that can be represented in both the knowledge graph and text, such as Wikipedia anchors and entity names (WANG et al., 2014) or entity descriptions (ZHONG et al., 2015). The goal of aligning these embeddings is to better capture the relationships between entities and concepts and to enable reasoning about complex information.

In Wang et al. (2014), the authors generate knowledge embeddings by using Freebase as KG and generate word embeddings by using Wikipedia pages. They used two alignment models to align the two types of embeddings. The first model replaced Wikipedia anchors with

their corresponding entities in Freebase, while the second model generated new triples in the knowledge graph. According to the authors, each alignment model has its own loss functions. They also noted that the first alignment model is integrated into the knowledge model, while the second alignment model is integrated into the word model.

In Zhong et al. (2015), the entity descriptions present in KGs are used to align the knowledge and word embeddings. Their knowledge and word models are the same employed by Wang et al. (2014). Their alignment model, however, does not replace words by entities or create new triples. The authors define a conditional probability of predicting for each word w in the description of an entity e, denoted as $Pr(w|e)$ using the embedding representation of the entity d. Moreover, they also calculate the conditional probability of an entity e for a word w, i.e., $Pr(e|w)$. Similarly to $Pr(w|e)$, the authors employ the embedding representation of the word w to estimate $Pr(e|w)$.

As stated by Bojanowski et al. (2017), many word embedding techniques treat each word as a distinct token, neglecting the internal structure of words in the process of learning their representation. This limitation poses a challenge for using such techniques in morphologically rich languages. Additionally, since these techniques treat words as separate tokens, the learned representation of a word is dependent on its frequency in the training dataset. If a word does not appear in the dataset, it will not have a representation. To circumvent those limitations, Bojanowski et al. (2017) proposes fastText. In Joulin et al. (2017a), the authors employ several concepts like low-rank constraints and bag of n-gram characters to improve the model's efficiency.

FastText is designed to generate representations for n-grams of characters. It constructs word embeddings by concatenating the n-grams of the characters that form the word. This strategy enables the creation of embeddings for words that are infrequently or never encountered in the training dataset.

According to Joulin et al. (2017b), the approach used in FastText for generating word embeddings can also be extended to generate knowledge embeddings. The authors achieved this by treating entities and concepts as separate tokens and considering a triple as a sentence. Additionally, the implementation of FastText allows each entity or concept to have one or more textual property values associated with it, which are used in the traditional FastText approach by considering n-grams of characters. This enables the joint training of both word embeddings and knowledge embeddings using FastText.

FastText provides competitive results for both word embedding and knowledge embedding, although not the best ones, as demonstrated in (BOJANOWSKI et al., 2017; JOULIN et al., 2017a; JOULIN et al., 2017b). Additionally, the authors note that FastText's advantage lies in its ability to train embeddings quickly, providing better results than other methods with similar training times. Given these advantages and its capacity for training word and knowledge embeddings jointly, we opted to use FastText for training the embeddings in OPHELIA. Specifically, we used KG triples and entity abstracts as inputs to train knowledge embeddings and word embeddings jointly in the same vector space.

2.5 NEURAL NETWORKS FOR TEXT CLASSIFICATION

Neural networks, especially deep learning, are highly effective in analyzing complex data, such as image and text classification, by modeling intricate relationships. They are used in various fields, including text analysis, to extract features from text. Neural networks emulate the processing of the human brain with connected nodes arranged in layers that transform inputs and discover relationships between them. They can learn on their own, store data, apply knowledge to real-time events, perform multiple tasks simultaneously, and automate feature engineering with deep learning, which identifies correlated features for quick learning (STEIN; JAQUES; VALIATI, 2019).

In Chapter 3, we present and compare several text classification approaches that use embeddings to represent features. Several of those approaches use different neural networks in their classification processes, as shown in Table 2. Among the many neural network architectures employed by these approaches, we have chosen to work with two in this thesis: FeedForward Neural Networks and Capsule Neural Networks.

2.5.1 Feedforward Neural Network

A Feedforward Neural Network (FFNN) (BEBIS; GEORGIOPOULOS, 1994) is an architecture in which information flows in one direction, moving only forward, from the input layer through one or more hidden layers to the output layer. The network is called "feedforward" because the output of one layer is fed as input to the next layer without any feedback loops. The nodes in the input layer receive input data, such as images or text, and the output layer produces the network's final prediction or output. Figure 6 shows the structure of an FFNN.

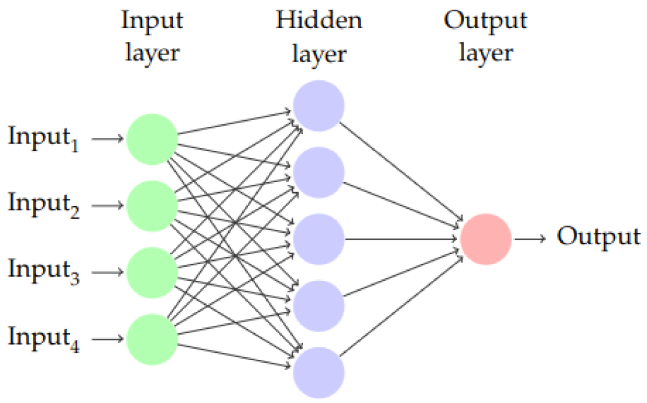

Figure 6 – Architecture of a FeedForward Neural Network (MONEDERO et al., 2014)

In a FFNN, each layer consists of multiple nodes (neurons) that perform mathematical computations on the input data and pass the result to the next layer. The input layer receives the input data, the hidden layers process it, and the output layer produces the final output. Each neuron in a layer is connected to all neurons in the previous layer, and the connections between

the neurons have associated weights. The weights are adjusted during training to optimize the network's performance for a specific task (DREYFUS, 2005).

FFNNs are trained using the backpropagation algorithm, which adjusts the weights of the connections between the neurons to minimize the error between the predicted output and the actual output, which helps the network to learn and improve its accuracy over time. The backpropagation algorithm works by propagating the error from the output layer back through the network and adjusting the weights of the connections based on the amount of error contributed by each connection (MONEDERO et al., 2014).

According to Bebis & Georgiopoulos (1994), the size of the input and output layers can be determined by the problem's dimensionality, while the size and number of the hidden layers are more empirical. However, the authors highlight that two hidden layers are enough to generalize most non-linear functions. In text classification approaches, an FFNN is mostly used to capture linear and non-linear relations between embeddings and other relevant features for named entity mention disambiguation, such as popularity and similarity scores.

2.5.2 Capsule Neural Networks

Capsule Neural Networks (CapsNets) were proposed by Hinton et al. (HINTON; KRIZHEVSKY; WANG, 2011) as an alternative to Convolutional Neural Networks(CNNs). CNNs have been successful in image processing tasks, but they have some limitations. One of the main limitations of CNNs is their inability to capture the spatial relationship between objects in an image. CNNs cannot handle pose, texture, deformation, and other object property variations. As a result of these limitations in CNNs, CapsNets were proposed as a promising solution. CapsNets have shown superior performance compared to CNNs for the aforementioned problems. (SABOUR; FROSST; HINTON, 2017; HINTON; SABOUR; FROSST, 2018).

In a CapsNet, a capsule is a group of neurons whose outputs represent different properties of the same entity. Each layer in a capsule network contains many capsules (SABOUR; FROSST; HINTON, 2017). The main idea behind capsule networks is to represent an entity (such as an object in an image) as a set of "capsules," which are groups of neurons that encode the various properties of the entity. Each capsule outputs a vector that represents the probability of the existence of the entity, as well as the various properties such as its orientation, size, color, and so on. These vectors are then used to reconstruct the entity by hierarchically combining them. Figure 7 represents simple a CapsNet with 3 layers.

Unlike CNNs, which are mainly focused on extracting local features from images, capsule networks are designed to extract both local and global features, as well as their relationships, in a more structured and interpretable way. Capsules are connected to each other in a way that allows them to learn how to represent complex relationships between features, such as spatial relationships between different parts of an object.

CapsNets introduce a new building block for deep learning that improves the modeling of hierarchical relationships within a neural network's internal knowledge representation.

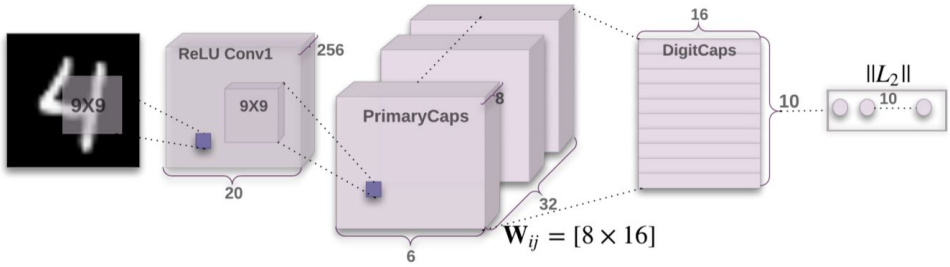

Figure 7 – Architecture of a CapsNet (SABOUR; FROSST; HINTON, 2017)

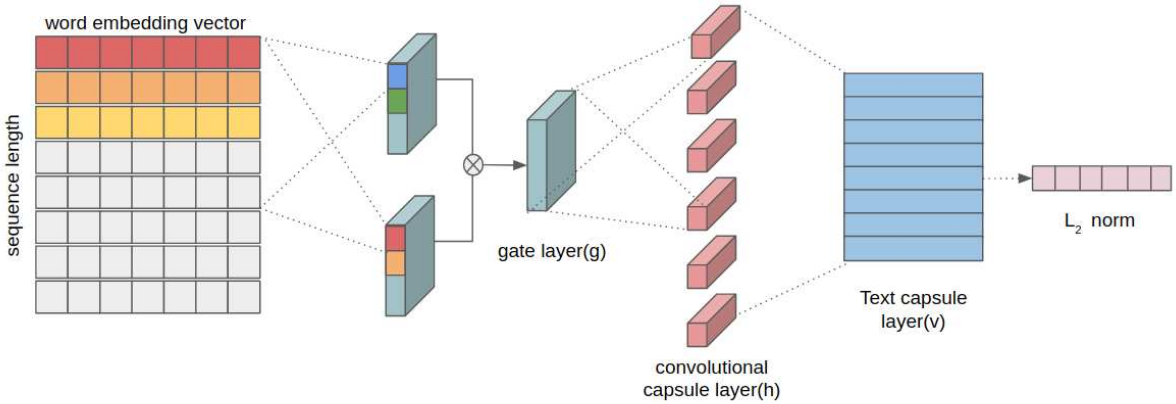

Figure 8 – A capsule network for text classification (KIM et al., 2020)

Capsules, the basic units of CapsNets, are clusters of neurons that exhibit equivariance by representing distinct attributes of the same feature as vectors instead of scalar values. This enables CapsNets to recognize entire objects by first detecting their individual components, making it possible to learn features of an image with deformations and varying viewing conditions (PATRICK et al., 2019; SABOUR; FROSST; HINTON, 2017).

Although they were first used for image classification, CapsNets can be used to exploit textual information in a more efficient manner in the areas of NLP and recommender systems (KATARYA; ARORA, 2019). For example, Kim et al. (2020) apply CapsNets to text classification, modifying them according to particular purposes. Figure 8 shows a capsule network structure for text classification. Each document passes a gate layer, a convolutional capsule layer, and a text capsule layer. In this regard, capsules are suitable to express a sentence or a document as a vector, where the spatial relationship between words and named entity mentions is relevant for understanding the textual contents. Other works also use CapsNets for text classification (ZHAO et al., 2019; KATARYA; ARORA, 2019). In the context of text classification, CapsNet can learn to recognize patterns within sentences or documents by encoding information about the spatial relationships between words and phrases. However, the practical application of CapsNet in text classification is still relatively limited compared to more established methods such as CNNs or recurrent neural networks (RNNs) like LSTM or GRU. The use of CapsNets in our own work is described in Section 4.3.

3 TEXT CLASSIFICATION USING EMBEDDINGS: SURVEY AND DIRECTIONS

This chapter is derived from our published article (COSTA; OLIVEIRA; FILETO, 2023), offering a comprehensive systematic review of text classification approaches that employ embeddings as feature representation. Systematic reviews are indispensable in the realm of science as they provide comprehensive and unbiased syntheses of existing research, empowering policymakers, researchers, and practitioners to make well-informed decisions grounded in the highest quality evidence. Consequently, these reviews help identify research gaps, directing researchers towards crucial unanswered questions, and propelling the advancement of knowledge within their respective domains. Section 3.1 describes the procedure adopted for the systematic bibliographic review. Section 3.2 analyzes semantics and embeddings in text classification, presenting key aspects of text classification approaches identified from existing works. Section 3.3 details the text classification approaches selected in our systematic bibliographic review. Section 3.4 presents a summary comparison of the text classification approaches and a discussion of these approaches. Section 3.5 evaluates the text classification approaches based on their results, metrics, and datasets used in their experiments.

3.1 BIBLIOGRAPHICAL REVIEW PROCEDURE

The steps of the methodology employed in the systematic bibliographical review are presented in Figure 9. The search for papers was done in Google Scholar[1], the ACM Digital Library[2], Science Direct[3], ACL Anthology [4] and, Scopus[5]. We chose these platforms because they gave the best results in preliminary searches for books and papers published in conference proceedings or journals. The search string used was (“Text Classification” OR “Document Classification”) AND “Embeddings”. We have noticed that this simple search string provides better results than a complex one, with the drawback that we had to filter more works in the following steps.

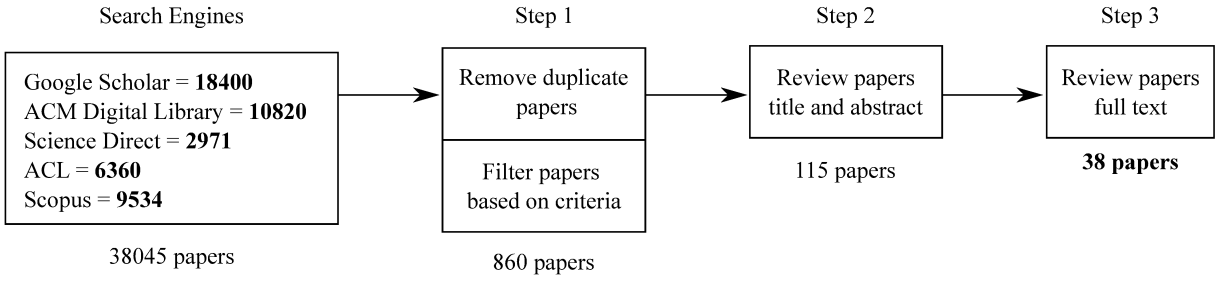

Figure 9 – Steps followed to retrieve the papers analyzed.

[1] https://scholar.google.com.br/
[2] https://dl.acm.org/
[3] https://www.sciencedirect.com/
[4] https://aclanthology.org/
[5] https://www.scopus.com/

We found 38045 articles. To reduce this number, in Step 1 we removed duplicates using Mendeley[6]. Then, we considered only articles that satisfied the following criteria: (i) peer-reviewed; (ii) published from 2015 onwards; (iii) written in English; (iv) text classification approaches that use embeddings. Step 1 yielded 860 articles. In Step 2, we manually analyzed the title and the abstract of the 860 articles resulting from Step 1, and removed those not related to text classification. Lastly, in Step 3, we reviewed the remaining 115 articles and selected 38 that use embeddings.

3.2 SEMANTICS AND EMBEDDINGS IN TEXT CLASSIFICATION

This section aims to draw how semantics and embeddings can contribute to text classification. First, Section 3.2.1 describes key aspects of text classification approaches that use embeddings and establish fundamental criteria to identify and analyze these approaches. Then, Section 3.2.2 gives an overview of the types of embeddings that can be used as features for text classification.

There are several ways to capture and represent semantics as features, and a variety of algorithms and techniques that can use these features in automatic approaches for text classification, as discussed in the following.

3.2.1 Key aspects of text classification approaches

Some key aspects for the comparative analysis of text classification approaches have been devised from our bibliographical review, and concrete examples of text classification like the one just presented (Figure 1). These aspects include the characteristics of the text classification problem at hand, the classification methods employed, what data inputs and data features are considered, and feature representation. These aspects are used to help distinguish, classify, and compare text classification approaches from the literature, besides providing insights about promising research directions. They can be described in more detail as follows:

Classification problem characteristics: While binary classification is the most common for texts, current industry needs are putting pressure to go beyond. Binary classification can already be challenging by itself, depending on the domain. However, text classification may happen in scenarios that involve multiple categories. For large sets of categories, a hierarchical structure is usually present. Taking advantage of such a structure during the learning and prediction processes defines what hierarchical classification is about. However, there are still some challenging issues, such as multi-class classification with many specific categories having subtle distinctive traits, unbalanced class distributions in the training datasets, and limitations in the labeled data available for training the classification model.

[6] https://www.mendeley.com/

Classification techniques: One of the most important issues in text classification is choosing the classification technique. Without a conceptual understanding of each technique, one may not effectively determine the most suitable one for a given application. Some of the most used and recommended machine learning techniques for text classification are naive Bayes, decision tree, support vector machine (SVM), and Neural networks. Nowadays, it is well known that neural networks, particularly deep learning approaches have achieved surpassing results on tasks such as image classification, face recognition, and several NLP tasks, including text classification. Nevertheless, some applications can be challenging, and the machine learning and deep learning areas are still evolving, with new alternative architectures arising frequently.

Data inputs and data features: Different text classification approaches can have distinct inputs. Some approaches are adapted to classify specific kinds of text documents (e.g., microblog posts, news articles, and academic papers). Thus, there are different domains with their peculiarities. For example, social media posts usually have short texts and present a more informal language with plenty of noise (typos, grammar errors, slang, acronyms, hashtags, etc.). Meanwhile, news articles and academic papers usually have longer documents with more formal text. In some cases, especially when context information is limited, as frequently happens with microblog posts, it is hard to grasp precise semantics from the texts.

Feature representation: Machine learning and deep learning methods, including those usually employed for text classification, require input features represented as fixed-length vectors of numbers. Techniques to extract features from text and represent them as vectors range from bare counting of word occurrences in a text document to techniques that measure word relevance, capture semantics, and reduce the dimensionality of the vector representations while keeping important properties. In this thesis, we focus on the use of embeddings for feature representation. In the following, we describe vector representations in general and the major kinds of embeddings that can be useful in text classification.

3.2.2 Embeddings as features for text classification

Since texts have no explicit features, much work has aimed to develop effective text representations (WU et al., 2018). Some feature representations are more sophisticated and carry more information than others, which impacts the performance of a NLP model (FIGUEIREDO et al., 2011). In addition, features extracted from text must be converted into a format that a machine can manipulate efficiently (GROSMAN et al., 2020). Thus, representation plays a vital role in many NLP-based tasks, such as text classification and clustering.

In conventional text classification, a document is usually represented based on the bag of words (BoW) model (ZHANG; JIN; ZHOU, 2010). This simplistic model represents each text document d as a vector v, whose dimensionality is the number of words of the vocabulary

V, and each numeric value $v[i](1 \leq i \leq |V|)$ quantifies the importance of a word $w_i \in V$ to the document d (e.g., word w_i frequency in the document d (TF), inverse document frequency for w_i in d (IDF), TF*IDF) (JONES, 1972; SCHUTZE; MANNING; RAGHAVAN, 2008). Consequently, the BoW model frequently leads to sparsity (due to words that do not appear in certain documents) and scalability problems (due to the size of the vocabulary V, which can include thousands of words). Furthermore, the BoW model ignores word order, word senses (which can be determined by neighboring words in the fine context where each word occurs), and semantic relations between senses (which can be represented in a knowledge graph, for example).

More recently, embeddings emerged as a means to circumvent these problems and improve text classification performance, among several other tasks. Embeddings encode complex data, such as words, word senses, or KG nodes and their relations in compact vectors (usually with a few hundred dimensions) that are suitable for efficient processing (CUI et al., 2018). Embeddings aim to represent features in continuous vector spaces, preferably low dimensional ones (BENGIO et al., 2003; LAI et al., 2016). Consequently, embeddings help to improve the results of text classification (among other NLP tasks) while maintaining the scalability of classification approaches for large amounts of texts (ALMEIDA; XEXÉO, 2019; BAKAROV, 2018). Ideally, an embedding should represent the data without loss of information, i.e., embeddings aim to preserve relevant or useful properties of the original data. In theory, a good embedding would produce exactly the same data when converted back to their original representation (NICKEL et al., 2016a). However, in practice, losses and distortions of the original data can occur. Thus, the goal is to preserve in the embedded representation what is relevant for some purpose (e.g., structural and/or semantic properties).

Usually, features that are alike, among other properties, keep a short distance between them in the embedded space. Figure 10 illustrates this in an example with word embeddings produced by the Word2Vec technique (MIKOLOV et al., 2013b), which is based on a neural network model. The Embedding Projector[7] was used to further reduce the dimensionality of the 200-dimensional Word2Vec embeddings and generate the 3-dimensional visualization presented in Figure 10. The embeddings of words with similar senses are close to each other. For instance, the embedding of the word *apple*, which can refer to the fruit or a corporation, as discussed in the text classification examples of Section 1.1, is close to those of both words *fruit* and *macintosh*.

Figure 11 presents a taxonomy of the major kinds of embeddings that have been or can be used to represent features potentially useful for text classification. *Text embeddings* are usually produced by applying non-supervised learning techniques to huge volumes of texts from sources like Wikipedia and news. Text embeddings can be *Context-independent* or *Context-dependent*. The former has a unique vector representation for each lexicon, while the latter can have distinct representations for each word occurrence, to capture sense variations in accor-

[7] https://projector.tensorflow.org/

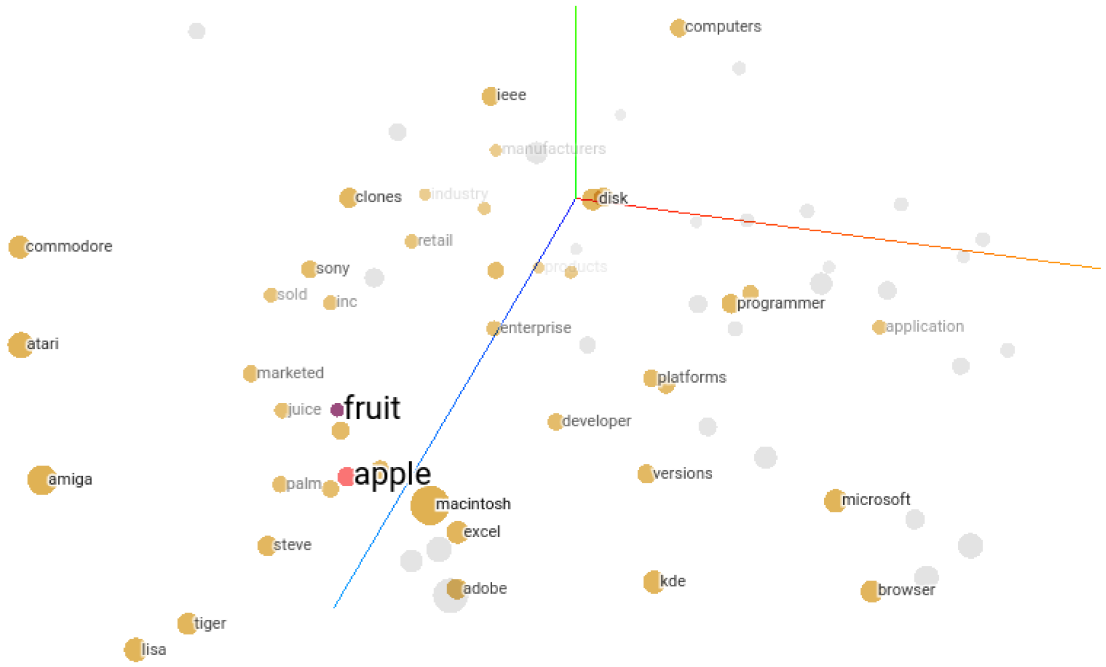

Figure 10 – Three-dimensional projection of Word2Vec embeddings of some words.

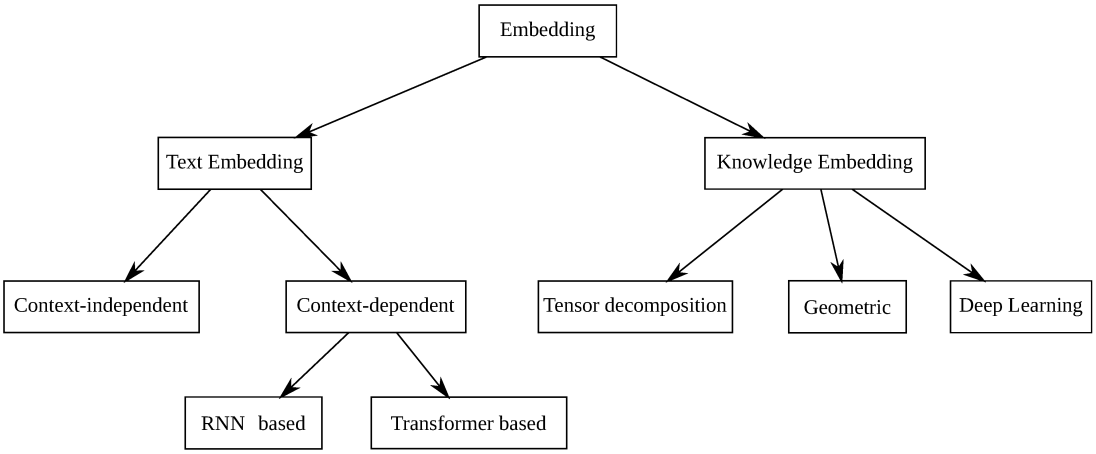

Figure 11 – Taxonomy of embeddings that can be useful for text classification

dance with the textual context. Consequently, context-dependent embeddings can distinguish word senses such as *apple (fruit)*, *Apple Inc.*, and *Apple Records*. Recently proposed context-dependent embeddings have yielded considerable performance gains over context-independent embeddings in several NLP tasks, including text classification. *Knowledge Embeddings*, on the other hand, codify knowledge such as nodes and relationships of a KG in representations based on vectors and/or tensors. They can be further classified according to the techniques used to generate them, such as *Tensor decomposition*, *Geometric*, and *Deep Learning*.

Currently, the most competitive approaches for text classification employ context-dependent embeddings of text, especially word embeddings. Their current popularity in text classification approaches is highlighted in Section 3.4. However, despite their wide popularity, text embeddings, even context-dependent ones, do not precisely represent semantic relations between concepts and entities as KGs do. In fact, the scientific community in the area of NLP

is currently discussing what exactly textual embeddings capture from the texts used for training them, and if additional knowledge from external sources is necessary to properly solve some NLP tasks, as human beings do in some situations (WANG; CUI; ZHANG, 2019). Thus, knowledge embeddings can theoretically become another alternative to represent useful features to classify texts with little contextual information, for example. These embeddings provide a suitable compact representation of facts present in a KG, about things that can be just mentioned in a text. Nevertheless, the potential of KGs and knowledge embeddings for leveraging text classification remains an open research question.

3.2.3 Effective Combinations of Classification Techniques with Embeddings

The vast majority of the text classification approaches found in the literature employ supervised learning, including generative (e.g. naive Bayes) and discriminative (e.g. SVM, random forest, neural networks) techniques. Figure 12 shows the most effective combinations of classification techniques (white rectangles) and embeddings (blue rounded rectangles) that we have identified in our review of the text classification literature.

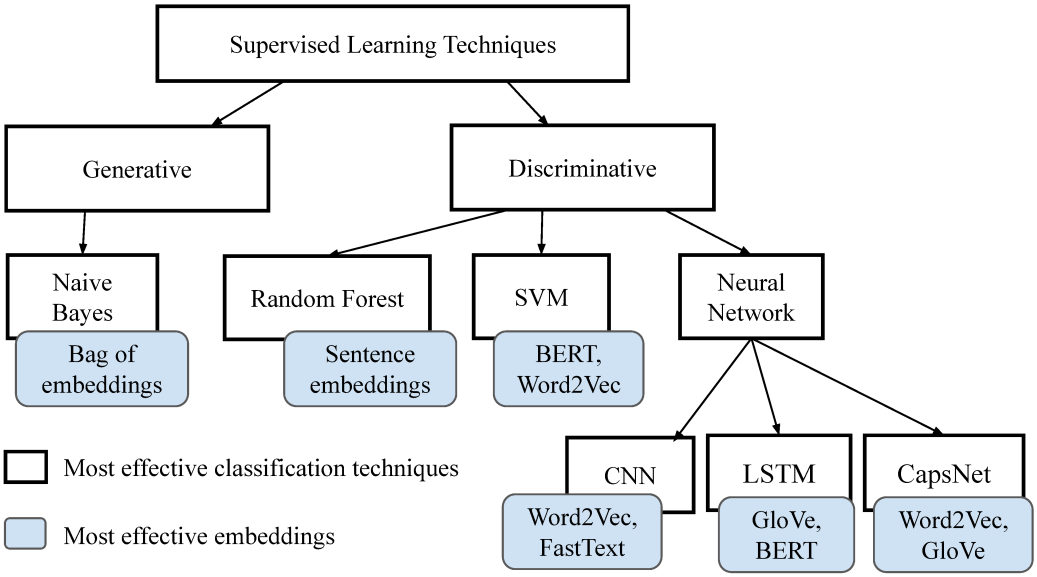

Figure 12 – Most effective combinations of classification techniques and embeddings identified in the text classification literature.

The classification techniques in the leaves of the tree presented in Figure 12 achieved the best performance and accuracy above 90% by using the embeddings indicated below the respective leaves as features to classify texts from at least one dataset listed in Table 6. These best results were obtained by using naive Bayes with bag-of-embeddings, random forest with sentence embeddings, SVM with BERT and Word2Vec; CNN with Word2Vec and FastText, LSTM with BERT and GloVe, and Caps net with Word2Vec and GloVe.

Section 3.3 describes and discusses some details of the text classification approaches selected from the literature and analyzed to draw these conclusions. Then, in Section 3.5,

Table 2 lists the classification techniques and embeddings employed by each approach, besides the domain of the texts they are intended to, while Tables 6 and 7 list the performance measures of these approaches on the datasets they were evaluated with, according to the results available in the papers that introduced the respective approaches.

3.3 CURRENT TEXT CLASSIFICATION APPROACHES USING EMBEDDINGS

This section reviews text classification approaches that explicitly use embeddings of any kind as features for text classification. The reviewed works were collected in an extensive and systematic review, as explained in Section 3.1. They are described and analyzed in the following subsections, according to the kinds of embeddings employed and their chronological evolution. Their classification techniques, premises, purposes and achieved performance are presented and discussed. The first 3 subsections refer to the embedding models that are most prominent in the text classification approaches that we have analyzed, namely Word2Vec, Glove, and BERT. The following subsection examines works that use less popular embedding models. Works using more than one embedding model are described in the subsection referring to the model that we consider paramount in the proposed solution if we identify such a model. Otherwise, that work is discussed in a final subsection about text classification approaches that use diverse embeddings or are produced by combining models.

Although most text classification proposals providing high performance are intended for binary classification, a considerable number of recent works from the literature tackle multi-class and multi-label classification. Thus, we consider in our review selected proposals for any of these classification problem variations. However, to keep the scope manageable, this work does not address particular classification problems and techniques, such as multi-level text categorization (TRIPATHI; OAKES; WERMTER, 2015; GUO et al., 2016; AGGARWAL; SINGH; GUPTA, 2018) and micro-word-based approaches (AL-ANZI; ABUZEINA, 2017), among other promising ones that have been investigated in the literature.

3.3.1 Approaches using Word2Vec

Word2Vec (MIKOLOV et al., 2013b) is a statistical approach for learning word embeddings from a text corpus, developed by Tomal Mikolov with the intent to make neural network-based training more efficient. It has become a benchmark for developing pre-trained context-independent word embeddings, and is one of the most used for text classification approaches.

Lai et al. 2015 (LAI et al., 2015) present a recurrent convolutional neural network for text classification. Their model captures contextual information and builds feature representations using the recurrent structure, which may introduce less noise than traditional window-based neural networks. In addition, it uses a max-pooling layer that automatically judges which words play critical roles in text classification. The pre-training of word embeddings fed to the neural network uses the Skip-gram model. The experimental results show that the neural net-

work approaches outperform the traditional methods for all four tested datasets. The highest accuracy rate achieved was 96.49% on the 20NewsGroup dataset. The authors affirm that it was possible due to the combination of the recurrent structure and the max-pooling layer, and the advantages of both neural models, recurrent and convolutional. The first is effective in capturing contextual information, and the latter may better capture the semantics of texts compared to recursive or recurrent neural networks.

Lenc et al. 2017 (LENC; KRÁL, 2017) investigated the use of word embeddings for representing long texts in multi-label classification. The embeddings were used in three convolutional neural network topologies. The authors analyzed the semantic similarity of the embedding vectors learned during the network training and compared them with the standard word2vec vectors. The precision score achieved was 91.03%. It was observed similar results for both variants learned by CNN. They concluded that initialization of the embeddings with Word2Vec pre-trained vectors does not play an essential role in text classification. However, learning word vectors is crucial to obtain good results.

Zhao et al. 2018 (ZHAO et al., 2018) exploited capsule networks with dynamic routing, a variant of the capsule networks introduced in (SABOUR; FROSST; HINTON, 2017). They proposed three strategies to stabilize the dynamic routing process to alleviate the disturbance of some noise capsules, which may contain background information or have not been successfully trained. The basic idea of dynamic routing is to construct a nonlinear map in an iterative manner, ensuring that the output of each capsule is sent to an appropriate parent in the subsequent layer. They also show that capsule networks significantly improve classification over competitors, especially for doing multi-label instead of single-label classification. Their model substantially and consistently outperforms simpler deep neural networks such as LSTM, Bi-LSTM, and CNN-rand by a noticeable margin on all the experimental datasets. Their capsule network model also achieves competitive results against more sophisticated deep learning models such as LR-LSTM, Tree-LSTM, VC-CNN, and CL-CNN. The approach achieved 93.8% of the accuracy score. However, though the capsule network exhibits high performance in single-label text classification, multi-label text classification is a more challenging problem because more training is required to cover the whole label space. Some works (LAI et al., 2015; LIU et al., 2018; PAN et al., 2019; PITTARAS et al., 2021) that employ Word2Vec for text classification focus on semantic characteristics and contextual information.

Guo et al. 2018 (GUO et al., 2019) proposed a novel term weighting scheme to be combined with word embeddings to enhance the classification performance of CNNs, considering the fact that one term generally has different importance in documents with distinct class labels. In this scheme, multiple weights are assigned to each term, and these weights are applied separately to the embedding of each word. The weighted word embeddings generated are fed to a multi-channel CNN model with the above-obtained weight vectors to execute classification. The input of each channel is obtained by applying the weight vectors corresponding to each class to the word embedding matrix, while in the original Kim's CNN model (KIM, 2014), the inputs for two channels are two-word embedding matrices obtained by CNN-static and CNN-

non-static schemes, respectively. Their experiments with publicly available Word2Vec vectors calculated the weights with training data only and trained the multi-channel TextCNN model using mini-batches, with each batch consisting of 50 documents. By comparing the novel method with several other baseline methods with five benchmark data sets, the results manifest that the classification accuracy of the proposed method exceeds that of other methods by a considerable margin. The highest accuracy rate achieved was 95.6% on the Subjectivity dataset.

Liu et al. 2018 (LIU et al., 2018) proposed the Task-oriented Word Embedding method (ToWE), which was specially designed to capture semantic and task-specific features of words for text classification. It introduces the function-aware component, which highlights the word's functional attributes in the embedding space by regularizing the distribution of words to have a precise classification boundary. This proposal uses Word2Vec to model the context information and log-linear models to produce word embeddings. The authors conclude that their method performed better than the compared methods. In particular, the ToWE-SG version of the method significantly outperforms the other baselines on the tested datasets, achieving 90.8% of the accuracy score in IMDB dataset.

Differently from (LIU et al., 2018), Pan et al. 2019 (PAN et al., 2019) proposed the Simple Word Embedding-based Model (SWEM) for text classification, which uses a modified hierarchical pooling strategy for simple word embedding in the few-shot transfer learning style. The model leverages and transfers knowledge obtained from some source domains to recognize and classify unseen text sequences with just a handful of support examples in the target problem domain. Extensive experiments using SWEM with Word2Vec and SVM to classify texts in English and Chinese from five datasets, reaching 92.1% of the accuracy score in DBpedia dataset, demonstrated that SWEM with parameter-free pooling operations is able to abstract and represent textual semantics useful to improve the results of text classification.

Meanwhile, Shi et al. 2019 (SHI; WANG; LI, 2019) consider the fusion of artificial neural network models to overcome the limitations of using only textCNN or LSTM for classifying news. They proposed a C-LSTM with word embedding model to obtain more accurate features considering the context information. This model extracts local features from the document through textCNN and global specialties referring to the whole document through LSTM. To preserve the original document better, the model integrates the word embedding into the fusion layer. As the most classical model, textCNN acquired a high classification accuracy and fast speed. However, the proposed method C-LSTM achieved the best result in experiments applied to the Chinese news dataset, achieving an accuracy rate of 86.53%.

Pittarras et al. 2021 (PITTARAS et al., 2021) applied semantic augmentation to enhance semantics of the inputs of deep neural networks used for text classification. Their proposal selects frequency-based semantic information from the WordNet semantic graph and fuses it with deep neural embeddings. For each word, it extracts semantic information from an appropriate source of existing knowledge, such as a semantic graph. It generates a vector for each word and represents the whole text as a fusion of the word semantic vectors. This approach achieves the best average performance when using raw concept frequencies, selecting

the first retrieved concept per word (basic strategy), and concatenating the resulting vector to the Word2Vec embedding. The highest accuracy value reached was 97.6% in the BBC corpus. They concluded that the way of introducing semantic information to the model affected training and the performance of the learned model.

Liu et al. 2021 (LIU et al., 2021) presented a model for multi-label text classification with many classes, called joint learning from Label Embedding and Label Correlation (LELC). It is based on the multi-layer attention and label correlation. LELC considers both the co-occurring label matrix and the label correlation matrix to exploit the potential label information and label correlation. Firstly, it uses a bidirectional gated recurrent unit network (BIGRU) to extract basic features, capture text contents information, and sequence information at the same time, and then apply the multi-layer attention framework to facilitate selecting valid features related to labels. Then, the label correlation matrix is taken into account in the process of performing label space dimension reduction (LSDR), which is fundamental for multi-label learning to simplify the process of model learning. At last, deep canonical correlation analysis technology was used to couple features and latent space in an end-to-end pattern. Experimental results in real-world datasets demonstrate the effectiveness of the model, achieving 92.22% of the Macro F1 score.

Finally, Gallo et al. 2021 (GALLO et al., 2021) presented a method that maps text features to image vectors to take advantage of image neural models to classify text documents. They convert each text document into an encoded image by using word embeddings. Analogous to word embedding models that can produce similar word embeddings for words occurring in equivalent contexts, their approach exploits this property to transform a text document into a sequence of colors (visual embedding), obtaining an encoded image that keeps similarity with the image encoding of similar documents. Their approach computes the Word2Vec word embedding of a text document, quantizes the embedding, and arranges it into a 2D visual representation as an RGB image. As a result, the method achieved competitive performance on well-known benchmark text classification datasets. The evaluated metric was percentage error, achieving 1.07% in the DBpedia dataset.

Notice that, most of the text classification approaches previously discussed are based on deep neural networks. The capsule network (SABOUR; FROSST; HINTON, 2017) has exhibited a better transferring capability than conventional deep neural networks, namely recurrent neural networks (RNNs) and convolutional neural networks (CNNs). On the other hand, just a few works using Word2Vec specifically addressed multi-label text classification (LENC; KRÁL, 2017; LIU et al., 2021).

3.3.2 Approaches using GloVe

Global Vectors for word representation (GloVe) (PENNINGTON; SOCHER; MANNING, 2014) uses the matrix factorization technique to embed the word-context matrix. This large and usually sparse matrix maintains for each word w of the vocabulary V a measure of the

co-occurrences of each other word of $V \setminus \{w\}$ in the textual context (close neighborhood consisting of a few words on each side of some occurrence of w) in some corpus. The idea behind this matrix is to derive the relationship between words from statistics of their co-occurrences in local text contexts. The approach is similar to the Word2Vec method, where each word is presented by a high-dimension vector and trained based on the surrounding words over a huge corpus.

GloVe has been successfully used in text classification approaches (ZHANG; LERTVITTAYAKUMJORN; GUO, 2019; CHALKIDIS et al., 2019; KIM et al., 2020; MOREO; ESULI; SEBASTIANI, 2021). Chalkidis et al. 2019 (CHALKIDIS et al., 2019) released a new dataset of 57k legislative documents from EUR-LEX, annotated with approximately 4.3k EUROVOC labels, intended to zero-shot learning, but applied in large multi-label text classification. They reported several experiments with different neural classifiers using pre-trained GloVe embeddings. They found out that a BIGRU with label-wise attention performs better than some state-of-the-art methods. They also investigated which zones of the documents are more informative on the dataset EURLEX57K, showing that considering only the title and recitals of each document leads to almost the same performance as viewing the entire document. The approach achieved 69.8% of the Macro F1 score. One major limitation of the investigated methods is that they are unsuitable for extreme multi-label text classification, with hundreds of thousands of labels.

Zhang et al. 2019 (ZHANG; LERTVITTAYAKUMJORN; GUO, 2019) proposed a novel CNN-based two-phase framework together with data augmentation and feature augmentation for recognizing text documents of classes that have never been seen in the learning stage (the so-called zero-shot text classification). It applies GloVe vectors as word embeddings. In fact, four kinds of semantically rich features (word embeddings, class descriptions, class hierarchy, and a general knowledge graph) are incorporated into the proposed framework to deal with instances of unseen classes effectively. The approach achieved 85.2% of the accuracy score. Therefore, the experiments show that data augmentation by topic translation improved the accuracy in detecting instances from unseen classes. In contrast, feature augmentation enabled knowledge transfer from seen to unseen classes for zero-shot learning.

Kim et al. 2020 (KIM et al., 2020) presented an application of capsule networks to the text classification domain and suggested utilizing a static routing variant to reduce the computational complexity of dynamic routing. Capsules consider the spatial relationships between entities and learn these relationships via dynamic routing. For this characteristic, capsules applied to text classification had advantages over the tested convolutional neural networks. The experimental results indicated that the achieved accuracy of 94.8% from the static-routing model is higher than that of the dynamic-routing model. Furthermore, the proposed model results were comparable results to those of initial studies regarding capsule network-based text classification.

Moreo et al. 2021 (MOREO; ESULI; SEBASTIANI, 2021) proposed word-class embeddings (WCEs), i.e., distributed representations of words specifically designed for multi-class text classification. They showed that, when concatenated to pre-trained word embeddings,

WCEs substantially facilitate the training of deep-learning models for multi-class classification by topic. They also showed empirical evidence that WCEs consistently improve multi-class classification accuracy, using six popular neural architectures and six widely used and publicly available datasets for multi-class text classification. Their method does not involve any optimization procedure but operates directly on the co-occurrence counters. The approach achieved 73.1% of the F1 score in the Ohsumed dataset.

According to the analyzed works, it is possible to observe different classification types as multi-class and multi-label text classification. In addition, distinct metrics(Macro F1, F1, Accuracy) were used to evaluate method performances in varied classification techniques. Therefore, the capsule network(SABOUR; FROSST; HINTON, 2017) achieved better performance in terms of accuracy.

3.3.3 Approaches using BERT

The BERT (Bidirectional Encoder Representations from Transformers) (DEVLIN et al., 2019) is a pre-trained neural network model used for natural language understanding tasks. It captures contextualized word embeddings by considering the entire sentence context, both left and right, of a given word. BERT's bidirectional approach makes it particularly effective at capturing the nuances of language, and it has become a cornerstone in various natural language processing applications. BERT is a big neural network based on the transformer architecture with a huge number of parameters that can range from 100 million to over 300 million. Several recent works have further studied and improved the BERT objectives and architecture and used it for different purposes. Some works applied it to text classification (CAI et al., 2020; MENG et al., 2020; LEE; LEE; YU, 2021; JIANG et al., 2021).

Cai et al. 2020 (CAI et al., 2020) proposed a hybrid BERT model that incorporates Label semantics via Adjustive attention (HBLA). It is a hybrid neural network model to take advantage of both label semantics and fine-grained text information to improve classification. The approach creates the label correlations graph based on adjacency similarity and encodes this graph to capture structure information and correlations among labels. The proposal also includes the design of a novel attention mechanism called adjustive attention to measure the semantic relation between word and label. BERT was used to capture the label-related discrimination information from each document and to obtain the implicit representation in each word's context. The experimental results achieved 90.6% of the precision score.

Meng et al. 2020 (MENG et al., 2020) presented the Label-Name-Only Text Classification (LOTClass) model to explore the potential of using only the name of each class to train classification models on unlabeled data, i.e., without using labeled documents. Pre-trained neural language models are used as general linguistic knowledge sources for category understanding and as representation learning models for document classification. This method associates semantically related words with the label names, finds category-indicative words, trains the model to predict their basic categories, and generalizes the model via self-training. The effec-

tiveness of LOTClass is assessed on four benchmarks. However, there are complex cases where label names are insufficient to build the correct classification model. The approach achieved 91.6% of the accuracy score in the Amazon Review dataset.

The Out-of-manifold Regularization in Contextual Embedding Space for Text Classification (OoMMix), proposed by Lee et al. 2021 (LEE; LEE; YU, 2021), is a new approach to find and regularize the remainder of the space to address the over-parameterization problem, referred to as out-of-manifold. The motivation is that the embeddings computed from the words only utilize a low-dimensional manifold, while a high-dimensional space is available for the model capacity. Therefore, OoMMix discovers the embeddings that are useful for the target task but cannot be accessed through the words. Precisely, they synthesize the out-of-manifold embeddings based on two embeddings obtained from actually observed words to utilize them for fine-tuning the network. A discriminator is trained to detect whether an input embedding is located inside the manifold or not, and simultaneously, a generator is optimized to produce new embeddings that can be easily identified as out-of-manifold by the discriminator. In the end, the fine-tuning on the synthesized out-of-manifold embeddings tightly regularizes the contextual embedding space of BERT. The experimental results achieved an accuracy score of 99.03% in DBpedia dataset.

Jiang et al. 2021 (JIANG et al., 2021) proposed the Light deep learning model (LightXML), a fine-tuning of the single transformer model with dynamic negative label sampling. To make LightXML robust in predicting, they proposed dynamic negative sampling based on these generative cooperative networks to recall and rank labels. With generative cooperative networks, the transformer model can be end-to-end fine-tuned in extreme multi-label classification, which makes the transformer model learn powerful text representation. The dynamic negative sampling allows label ranking part to learn from easy to hard and avoid overfitting, which can boost overall model performance. Experiments showed that LightXML outperforms state-of-the-art methods in five extreme multi-label datasets. Although their experiments showed good results, 96.77% of the accuracy score, their model has a much smaller size and lower computational complexity than current state-of-the-art methods.

The presented approaches achieved satisfactory results. Some of them applied neural networks as a classification technique. However, regardless of the method used, all approaches obtained accuracy above 90%. In conclusion, the use of BERT enables preeminent model performance over legacy methods and an ability to process larger amounts of text and language.

3.3.4 Approaches using Other Embeddings

Besides Wor2Vec, GloVe, and BERT, other text embeddings have been used in a few text classification approaches. These embedding models include FastText (BOJANOWSKI et al., 2017), Doc2Vec (LE; MIKOLOV, 2014), and region embedding (QIAO et al., 2018), among others. Sparse Local Embeddings for Extreme Classification (SLEEC) was proposed by Bhatia et al. 2015 (BHATIA et al., 2015) as an extreme multi-label classifier to address the problem of

learning a classifier that can automatically tag a data point with the most relevant subset of labels from a large label set. The SLEEC algorithm extends embedding methods. Its main technical contribution is a formulation for learning a small ensemble of local distance preserving embeddings that can accurately predict infrequently occurring labels. During prediction, SLEEC uses a k-nearest neighbor (kNN) classifier in the embedding space, thus leveraging on the preservation of nearest neighbors during training. The experimental results achieved 85.54% of the precision score in Wiki10 dataset.

Qiao et al. 2018 (QIAO et al., 2018) presented a new learning method and utilized distributed representations of n-grams, referred to as “region embeddings", for text classification. The regions in a document, in this case, were considered as fixed-length contiguous subsequences of tokens in the document. The approach focused on learning the representations of small text regions, which preserve the local internal structural information for text classification. For utilizing the word-specific influences of each word on its context words, a local context unit for each word is learned in addition to word embedding. They used the interactions between words and their local context based on word embeddings as well as the local context units to produce region embeddings. In addition to the analysis of region sizes, they further studied the influence of word embedding dimensions. The experimental results achieved 98.9% of the accuracy score in DBpedia dataset.

The works (ALY; REMUS; BIEMANN, 2019) and (HOSSAIN; HOQUE; SARKER, 2021) use FastText embeddings in their text classification models. Hossain et al. 2021 (HOSSAIN; HOQUE; SARKER, 2021) introduced a convolution neural network-based model using FastText embedding for text document classification of resource-constrained languages. A corpus of documents in a low-resource language, namely Bengali, was developed to assess the performance of the proposed model, and different hyperparameters of the CNN model were tuned for optimization and hence to achieve better classification results. Despite the challenge to develop an automatic text classification system for low-resource languages such as Bengali, with a scarcity of digital resources and benchmark corpora, evaluation results on test datasets showed improved performance of the proposed method compared to existing techniques, such as TF-IDF-SVM, Word2Vec-SVM, GloVe-SVM, and FastText-SVM. The highest accuracy score achieved was 96.85%.

Aly, Remus, and Biemann 2019 (ALY; REMUS; BIEMANN, 2019) applied simple shallow capsule networks for hierarchical multi-label text classification and showed that they can perform superior to other neural networks, such as CNN’s and LSTMs, and non-neural network architectures such as SVMs. Results of experiments with pre-trained FastText embeddings adjusted during training confirmed the hypothesis that capsule networks are especially advantageous for rare events and structurally diverse categories. The approach achieved 82.75% of the precision score. Therefore, the main benefit of capsules shown is their ability to encode information of each category separately, by associating each capsule with one category. Combining encoded features independently for each capsule enables capsule networks to handle label combinations better than previous approaches.

Pappas and Henderson (PAPPAS; HENDERSON, 2019) proposed a novel joint input-label embedding model for neural text classification that generalizes over existing input-label models and addressed their limitations while preserving high performance on seen and unseen labels. Models were evaluated on full-resource and low or zero-resource text classification of multilingual news and biomedical text with a large label set. Two nonlinear transformations address the need to capture complex label relationships with the same target joint space dimensionality. The experimental results achieved 79.85% in the F1 score. Therefore, the approach differs from others due to the ability of the model to capture complex input label relationships, controllable capacity, and training objective, which is based on cross-entropy loss.

Also, based on region embedding, Li and Ye 2020 (LI; YE, 2020) presented a text classification model combining region embedding and a RELSTM to form a hybrid neural network. The RELSTM first divides regions for text and then generates region embeddings. The input of the RELSTM model is a word index, where each word embedding is initialized and continuously adjusted during the training process. This model does not require pre-trained word embeddings, which simplifies the experimental process and facilitates the investigation of the impact of region size, the number of layers, and hidden nodes on the accuracy of the RELSTM. The highest accuracy score achieved was 97.7% in Paper Theme dataset.

Finally, Chang et al. 2020 (CHANG et al., 2020) proposed X-Transformer, a scalable approach to fine-tuning deep transformer models for text classification, using a pre-trained XL-Net (YANG et al., 2019). X-Transformer consists of a Semantic Label Indexing component, a Deep Neural Matching component, and an Ensemble Ranking component. The proposed method achieves new state-of-the-art results on four benchmark datasets. The highest precision score achieved was 96.7% in AmazonCat-13k. However, successfully applying transformer models to extreme multi-label problems remains an open challenge due to the vast output space and severe label sparsity issues.

3.3.5 Approaches using Embedding Combinations

Several approaches combine distinct models to create embeddings for text classification. Xu et al. 2016 (XU et al., 2016) proposed the Topic-based Skip-gram approach to learn topic-based word embeddings and two CNN architectures that use multiple word representations simultaneously for text classification. This approach uses latent dirichlet allocation (LDA) to capture precise topic-based word relationships and integrate them into distributed word embedding learning with a novel objective function. The approach results achieved 95% in the Macro F1 score in 20NewsGroup dataset.

Jin et al. 2016 (JIN et al., 2016) built a text classifier using a naive Bayes model with a bag-of-embeddings that extends the skip-gram model (MIKOLOV et al., 2013b) to incorporate context and word sense information. To better do this, they train multi-prototype target word embeddings, with one distinct vector trained for a word under each class, i.e., each word embedding is produced separately. The context vectors of words remain the same across different

classes. Compared with bag-of-word models, the bag-of-embeddings model exploits contextual information by deriving the probabilities of class-sensitive embedding vectors from their inner product with context words. The proposed model achieved 96.5% of the accuracy rate.

Most of the existing methods for multi-label classification learn a single linear parametrization using the entire training set. Hence, they fail to capture nonlinear intrinsic information in feature and label spaces, due to the exponential size of the output space. To overcome this, Kumar et al. 2018 (KUMAR et al., 2018) presented a new multi-label classification method called MLC-HMF, which learns piecewise-linear embeddings with a low-rank constraint on parametrization to capture nonlinear intrinsic relationships that exist in the original feature and label space. The approach considered accuracy as the evaluated metric, achieving 93%. Therefore, the experimental analysis provided evidence that hierarchical embedding can yield more accurate results for multi-label classification.

Wang et al. 2018 (WANG et al., 2018) investigated label embeddings for text representations and proposed the Label Embedding Attentive Model (LEAM) to improve text classification. LEAM was implemented by jointly embedding words and labels in the same latent space, and the text representations were constructed directly using text-label compatibility. They introduced an attention framework that measures the compatibility between embeddings of text sequences and labels. The attention is learned on a training set of labeled samples to ensure that, given a text sequence, the relevant words are weighted higher than the irrelevant ones. Their method maintains the interpretability of word embeddings and enjoys a built-in ability to leverage alternative sources of information, in addition to input text sequences. Experiments were done in 6 different datasets, achieving 99.02% of accuracy in DBpedia dataset.

Liu et al. 2019 (LIU et al., 2019) presented a distributional representation of words combined with their part-of-speech tags. This embedding model is a modification of the continuous bag-of-words model that predicts the current word based on the context (MIKOLOV et al., 2013b). They build a two-dimensional look-up table on the training set in the format $\langle word, part\_of\_speech \rangle$ to represent word features. It makes word embeddings more expressive and semantically rich, improving the performance of a Bayesian classifier. If a new category is added, the model does not need to recalculate the word embeddings for this category. The experimental results achieved 90.2% of the accuracy score in DBpedia dataset.

Sinoara et al. 2019 (SINOARA et al., 2019) proposed two models to represent document collections based on both words and word senses, having the objective of improving text classification performance through enriching text representations with semantics. They use word sense disambiguation tools and available pre-trained word and word sense models to construct embedded representations of documents. The proposed approach has the potential to be applied to documents written in several languages since it relies on the multilingual Babelnet KG and pre-trained word embeddings. Their representations are low-dimensional, which helps to speed up the learning and classification processes. Their experimental evaluation indicates that the use of the proposed representations provides stable classifiers with reliable quantitative results, especially in semantically complex classification scenarios.

Le and Mikolov (MIKOLOV et al., 2013b) introduced Word2Vec and the doc2vec method for learning embeddings of phrases. They inspired embeddings for coarser textual granularities (LE; MIKOLOV, 2014) and their application to text classification. For example, (AUBAID; MISHRA, 2020) propose a rule-based approach using doc2vec embeddings in text classification. Their work investigates how varying rule-based classification and embeddings of distinct text granularities influence performance. Their document vector rule-based (D2vecRule) proposal, tested on the datasets Reuters-21578 and 20 Newsgroups, achieved 90.72% of accuracy in the first dataset, and good results according to the F-measures and implementation time metrics.

Gupta et al. 2020 (GUPTA et al., 2020) proposed a novel document representation technique, Sparse Composite Document Vector Multi-Sense (SCDV-MS), extended from SCDV to consider the multi-sense nature of words. SCDV-MS utilizes multi-sense word embeddings and learns a lower-dimensional manifold. Experiments on multiple real-world datasets showed that SCDV-MS embeddings outperform previous state-of-the-art embeddings on multi-class and multi-label text categorization tasks. The accuracy score achieved was 86.19%. Furthermore, comparing the results, SCDV-MS embeddings proved efficient in time and space complexity for textual classification.

Bounabi et al. 2020 (BOUNABI; MOUTAOUAKIL; SATORI, 2020) presented a study that allows understanding the advantages of Doc2vec and profit from them in neural embedding applications. They use Doc2vec to produce vector inputs for machine learning models through the variant Paragraph Vector-Distributed Memory (PV-DM). Then, it is incorporated into hybrid ML methods to improve classification quality. Different parameters control the effectiveness of the document representation. Experimental results prove that the architecture based on the PV-DM version with the average method, plus an optimal epoch number and minimal vector size, has a positive impact on text classification, achieving 99.1% of accuracy.

Hu et al. 2020 (HU et al., 2020) proposed an enhanced word embedding method that introduces a unique sentence reorganization technology to rewrite all the sentences in the original training corpus. Then, the original and the reorganized corpora are merged in a training corpus of the distributed word embedding model to solve the coexistence problem of words and phrases in the same vector space. The advantage of this method is that it can incorporate phrase features into the original vector space to form a hybrid distributed embedded structure where words and phrases coexist. In addition, it does not need any additional training corpus. Consequently, it can also alleviate the problem of insufficient word embedding corpus for training. The experimental results achieved 97.23% of the accuracy score in R52 dataset.

Liu et al. 2021 (LIU; WANG; REN, 2021) proposed a Label-Embedding Bidirectional Attentive model to improve the performance of the BERT text classification framework, extending this framework with label embedding and bidirectional attention (Text-to-Label Attention and Label-to-Text Attention). Therefore, it was built upon BERT, whose outputs are the sequence-level text representation and the token-level text representation. The bidirectional attention mechanism was proposed to integrate information between label embedding, sequence-

level text representation, and token-level text representation. Experimental results demonstrated the effectiveness of their proposal, achieving 94.4% of the Macro F1 score.

Zhang et al. 2021 (ZHANG; YAMANA, 2021) proposed an approach to incorporate knowledge about class labels into text classification models. In this approach, label-related knowledge is represented by keywords that users can customize. The relatedness between each word in the text sequence and hidden knowledge, such as keywords, is calculated and concatenated with the original model's information. Their proposal showed to be capable of understanding the relationship between sequences and labels, performing well on datasets with many classes. The highest accuracy score achieved was 94.72% in AG News dataset.

Finally, (SARASWAT; ABHISHEK; KUMAR, 2021) proposed the Language Agnostic Sentence Representations (LASER), which uses a single encoder to generate an embedded vector per sentence in any language. This approach focuses on purpose classification in different speeches, given that the model is trained only in one language. Several machine learning models were developed, and their outputs were compared to understand how zero-shot learning (ZSL) works. The highest accuracy score achieved in the experiments was 93%. The approach is grounded on the idea that the sentence embedding of two sentences of different languages having the same sense must be similar. Therefore, the effectiveness of this model can be tested when two different terminologies are used in the same sentences, i.e., writing words of one language in the alphabets of a second language or when there are two different sets of alphabets belonging to two different languages in the same sentence.

3.4 COMPARATIVE ANALYSIS OF TEXT CLASSIFICATION APPROACHES

This section presents a comparison summary of the approaches detailed in Section 3.3 that use embeddings as feature representations and our consideration of the state-of-the-art text classification approaches. Table 2 summarizes the text classification approaches related to this thesis. They are listed in chronological order, presenting the columns according to relevant aspects of text classification described in Section 3.2.1. The column *Domain* refers to the source and nature of the classified text documents. The *Method* employed by each work is indicated in the third column. Lastly, the column *Feature Representation* denotes the kind of feature representation used to classify the texts. Other methods and tools used to generate features or preprocess the data are not shown in this table because they are outside the scope of this thesis.

The column *Domain* of Table 2 allows one to perceive that most of the works are multi-domain, i.e., they are intended to work, trained, and evaluated with text from different domains. Furthermore, many works focus only on news articles. By analyzing Table 2 with the work details described in Section 3.3, it is possible to notice that most corpora have long and formal texts. Moreover, several ones refer to everyday affairs. Such datasets are probably used because embeddings are usually trained with texts that also refer to everyday affairs. Lastly, the use of long corpora may be indicative that the current approaches require a significant amount of context to be able to correctly categorize the text documents. From the analyzed works,

Table 2 – Classification approaches using embeddings (COSTA; OLIVEIRA; FILETO, 2023)

Work	Domain	Classification Technique	Feature Representation
Lai et al. 2015 (LAI et al., 2015)	News article, Academic paper, Sentiment Analysis	RCNN	Word embeddings, Word2Vec
Bhatia et al. 2015 (BHATIA et al., 2015)	Multi-domain	SLEEC	Word embeddings
Xu et al. 2016 (XU et al., 2016)	News articles, Medical articles	CNN	Word embeddings
Jin et al. 2016 (JIN et al., 2016)	News articles	Naive Bayes	Word embeddings, Bag-of-embeddings
Lenc et al. 2017 (LENC; KRÁL, 2017)	News articles	CNN	Word embeddings, Word2Vec
Qiao et al. 2018 (QIAO et al., 2018)	News article, Reviews, All-purpose	Word-Context region embedding, Context-Word region embedding	Word embeddings, Region embedding
Liu et al. 2018 (LIU et al., 2018)	News article, Reviews, Academic paper, Sentiment analysis	ToWE	Word embeddings, Word2Vec
Zhao et al. 2018 (ZHAO et al., 2018)	News article, Reviews, Sentiment analysis	CapsNet	Word embeddings, Word2Vec
Kumar et al. 2018 (KUMAR et al., 2018)	Sentiment analysis, Medical article, News article	MLC-HMF	Word embeddings
Wang et al. 2018 (WANG et al., 2018)	News article, All-purpose	LEAM	Label-Embeddings, Glove
Aly et al. 2019 (ALY; REMUS; BIEMANN, 2019)	Book blurbs, Academic articles	CapsNet	Word embeddings, FastText
Zhang et al. 2019 (ZHANG; LERTVITTAYAKUMJORN; GUO, 2019)	News articles, All-purpose	CNN	Word embeddings, GloVe
Chalkidis et al. 2019 (CHALKIDIS et al., 2019)	Legislative text	CNN, GRU, HAN	Word embeddings, GloVe, BERT
Pappas et al. 2019 (PAPPAS; HENDERSON, 2019)	News articles, Biomedical question	HAN, MHAN	Word embeddings, Label-Embeddings
Pan et al. 2019 (PAN et al., 2019)	News article, All-purpose	SVM	Word embeddings/ Word2Vec
Liu et al. 2019 (LIU et al., 2019)	News articles	LIBSVM	Word embeddings
Guo et al. 2019 (GUO et al., 2019)	Reviews, All-purpose	CNN	Word embeddings/ Word2Vec
Shi et al. 2019 (SHI; WANG; LI, 2019)	News articles	C-LSTM	Word embeddings/ Word2Vec
Sinoara et al. 2019 (SINOARA et al., 2019)	News article, Computer Science Technical Reports, Biomedical text, Sentiment Analysis	Naive Bayes, SMO, IMBHN	Word embeddings/ Knowledge embeddings/ Babel2Vec / Word2Vec
Cai et al. 2020 (CAI et al., 2020)	News articles, Academic paper	Bi-LSTM	Label embeddings/ BERT
Bounabi et al. 2020 (BOUNABI; MOUTAOUAKIL; SATORI, 2020)	Sport news	SVM, Logistic Function, FNN, Hybrid ML model	Word embeddings/ Doc2Vec
Hu et al. 2020 (HU et al., 2020)	Movie review, Academic paper	CNN	Word embeddings
Li et al. 2020 (LI; YE, 2020)	News article, Hotel review	LSTM	Region word embeddings
Aubaid et al. 2020 (AUBAID; MISHRA, 2020)	News articles	JRip (RIPPER), One Rule (OneR) ZeroR	Word embeddings/ Doc2Vec
Gupta et al. 2020 (GUPTA et al., 2020)	News articles	LinearSVM, Logistic regression, SCDV-MS	Word embeddings/ Doc2vec
Meng et al. 2020 (MENG et al., 2020)	News articles, Movie review, All-purpose	LOTClass	Word embeddings/ BERT
Chang et al. 2020 (CHANG et al., 2020)	All-purpose	X-Transformer	Word embeddings/ XLNet
Kim et al. 2020 (KIM et al., 2020)	News article, Reviews,	CapsNet	Word embeddings/ GloVe
Pittaras et al. 2021 (PITTARAS et al., 2021)	News article, Biomedical text	DNN	Word embeddings/ Word2Vec
Liu et al. 2021 (LIU et al., 2021)	News article, Medical article, Reviews, email text	LELC	Word embeddings/ Wor2Vec
Liu et al. 2021 (LIU; WANG; REN, 2021)	News article, Academic paper, Sentiment analysis	BERT classifier	Label embeddings/ BERT
Hossain et al. 2021 (HOSSAIN; HOQUE; SARKER, 2021)	News article	CNN	Word embeddings/ FastText
Saraswat et al. 2021 (SARASWAT; ABHISHEK; KUMAR, 2021)	Consumer's queries	Random Forest, SVM and Multilayer Perceptron model	Sentence embeddings
Gallo et al. 2021 (GALLO et al., 2021)	News article, Sentiment analysis, All-purpose	CNN	Word embeddings/ Word2Vec
Lee et al. 2021 (LEE; LEE; YU, 2021)	News article, Reviews, All-purpose	SVM, BERT	Contextual embeddings/ Bert
Zhang et al. 2021 (ZHANG; YAMANA, 2021)	News article, Movie review	Bi-LSTM	Word embeddings/ Knowledge embeddings/ GloVe, BERT
Jiang et al. 2021 (JIANG et al., 2021)	Multi-domain	Generative cooperative networks	Label embeddings/ BERT, XLNet, RoBERTa
Moreo et al. 2021 (MOREO; ESULI; SEBASTIANI, 2021)	News article, Biomedical text	SVM, BERT, TCNN, LEAM, FastText classifier, CNN, LSTM, ATTN	Word embeddings/ GloVe

only (SARASWAT; ABHISHEK; KUMAR, 2021) tackles short documents, to categorize consumer's queries. The classification of short documents, be they informal (like social media posts) or not, is a challenging research theme.

Regarding the column *Method*, notice that most works use neural networks, with varying architectures, as the classifier. This happens because neural networks, and particularly deep learning, have surpassed the performance of machine learning algorithms on several tasks, including text classification (STEIN; JAQUES; VALIATI, 2019). Deep learning can execute featuring engineering on its own and promote fast learning. Although it is not explicit in Table 2, most recent works employ deep neural networks, a hot topic in several research domains nowadays due to their superior performance and current availability of the necessary hardware. As the approaches employ a myriad of neural network architectures, which are still evolving fast, we prefer not to draw any conclusion about the most used and suitable architectures yet. However, it is possible to notice that several approaches employ neural network architectures

that consider, in some way, the order in which the words occur.

The chronological order of the proposals in Table 2 allows noting, in its last column, *Feature Representation*, that the embedding approaches are shifting from context-independent word representations, like Word2Vec, to approaches that use contextual embeddings, based, for example, on the transformer technology, as BERT. However, Word2Vec is still the most used embedding, followed by Glove. This is an indication that these embeddings are mature enough to be used in several domains.

Table 3 provides links to the repositories of open-source code (when available) or to the GitHub profiles of the author(s) of the approaches considered in this thesis, to facilitate access to what was produced by each work. Section 3.5 provides a performance comparison summary of these approaches.

3.5 EVALUATION OF TEXT CLASSIFICATION APPROACHES

The works analyzed in this thesis use several datasets and performance metrics to evaluate their text classification approaches. Table 4 and Table 5 provide pointers to the datasets used. Table 4 presents links to their respective home pages when they are available. When no such link is unavailable, Table 5 provides a reference to the paper or challenge in which the dataset appears.

The most used metric to evaluate the performance of the approaches considered in this paper is the accuracy score. It is usually characterized in terms of error and is traditionally decomposed into bias (systematic error) and variance (random error) components. The next most used evaluation metric is by far the F1. It tolerates uneven class distributions better than accuracy. Therefore, depending on the dataset used to evaluate the text classification approach, the accuracy may provide a less reliable measure of the performance than F1. Table 6 presents the accuracy score of works analyzed in this thesis (listed in the first column of the table, by the chronological order of their publications) that use this metric to evaluate performance on distinct datasets (listed in alphabetic order in the second line of the table header). Table 7 summarizes the performance of the works analyzed in this work that, instead of the standard accuracy score, use other metrics (listed in the first column of the table) to evaluate their approaches. The highest performance value (accuracy, F1, micro F1, macro F1, precision, recall) achieved by one of the classification methods on each dataset is highlighted in **bold** to allow the reader to better follow and understand the results presented in Tables 6 and 7. It is important to elucidate that the performance scores presented in these tables are taken from the papers that introduced the respective text classification approaches.

3.5.1 Directions for Text Classification Research

Recent advancements in feature representation and deep learning have propitiated significant progress in text classification, as discussed previously. These advancements include the

Table 3 – Links to the considered approaches. When available, a link to the source code repository is provided in the second column. Otherwise, the Github profile of each author is provided in the third column, on the respective full name ((COSTA; OLIVEIRA; FILETO, 2023)).

Work	Source code repository	Github of authors
Lai et al. 2015 (LAI et al., 2015)	https://github.com/roomylee/rcnn-text-classification	
Bhatia et al. 2015 (BHATIA et al., 2015)	https://github.com/xiaohan2012/sleec_python	
Xu et al. 2016 (XU et al., 2016)	https://github.com/HaotianMXu/Multimodal-CNNs	Haotian Xu
Jin et al. 2016 (JIN et al., 2016)		
Lenc et al. 2017 (LENC; KRÁL, 2017)		
Qiao et al. 2018 (QIAO et al., 2018)	https://github.com/schelotto/Region_Embedding_Text_Classification_Pytorch	
Liu et al. 2018 (LIU et al., 2018)	https://github.com/qianliu0708/ToWE	Qian Liu
Zhao et al. 2018 (ZHAO et al., 2018)	https://github.com/andyweizhao/capsule_text_classification	Wei Zhao
Kumar et al. 2018 (KUMAR et al., 2018)		
Wang et al. 2018 (WANG et al., 2018)	https://github.com/guoyinwang/LEAM	Guoyin Wang
Aly et al. 2019 (ALY; REMUS; BIEMANN, 2019)	https://github.com/uhh-lt/BlurbGenreCollection-HMC	Rami Aly
Zhang et al. 2019 (ZHANG; LERTVITTAYAKUMJORN; GUO, 2019)	https://github.com/JingqingZ/KG4ZeroShotText	Jingqing Zhang
Chalkidis et al. 2019 (CHALKIDIS et al., 2019)	https://github.com/iliaschalkidis/lmtc-eurlex57k	Ilias Chalkidis
Pappas et al. 2019 (PAPPAS; HENDERSON, 2019)	https://github.com/idiap/gile	Nikolaos Pappas
Pan et al. 2019 (PAN et al., 2019)		
Liu et al. 2019 (LIU et al., 2019)		
Guo et al. 2019 (GUO et al., 2019)		
Shi et al. 2019 (SHI; WANG; LI, 2019)		
Sinoara et al. 2019 (SINOARA et al., 2019)		
Cai et al. 2020 (CAI et al., 2020)		
Bounabi et al. 2020 (BOUNABI; MOUTAOUAKIL; SATORI, 2020)		
Hu et al. 2020 (HU et al., 2020)		
Li et al. 2020 (LI; YE, 2020)		
Aubaid et al. 2020 (AUBAID; MISHRA, 2020)		
Gupta et al. 2020 (GUPTA et al., 2020)	https://github.com/vgupta123/SCDV-MS	Vivek Gupta
Meng et al. 2020 (MENG et al., 2020)	https://github.com/yumeng5/LOTClass	Yu Meng
Chang et al. 2020 (CHANG et al., 2020)	https://github.com/OctoberChang/X-Transformer	Wei-Cheng Chang
Kim et al. 2020 (KIM et al., 2020)	https://github.com/TeamLab/text-capsule-network	Jaeyoung Kim
Pittaras et al. 2021 (PITTARAS et al., 2021)	https://github.com/npit/nlp-semantic-augmentation/tree/jnle	Nikiforos Pittaras
Liu et al. 2021 (LIU et al., 2021)		
Liu et al. 2021 (LIU; WANG; REN, 2021)		
Hossain et al. 2021 (HOSSAIN; HOQUE; SARKER, 2021)		
Saraswat et al. 2021 (SARASWAT; ABHISHEK; KUMAR, 2021)		
Gallo et al. 2021 (GALLO et al., 2021)	https://gitlab.com/nicolalandro/visual_word_embeddings	
Lee et al. 2021 (LEE; LEE; YU, 2021)	https://github.com/sh0416/oommix	Seonghyeon Lee
Zhang et al. 2021 (ZHANG; YAMANA, 2021)	https://github.com/HeroadZ/KiL	Zhang Cheng
Jiang et al. 2021 (JIANG et al., 2021)	http://github.com/kongds/LightXML	Ting Jiang
Moreo et al. 2021 (MOREO; ESULI; SEBASTIANI, 2021)	https://github.com/AlexMoreo/word-class-embeddings	Alejandro Moreo

Table 4 – Links to datasets used to train and evaluate text classification approaches ((COSTA; OLIVEIRA; FILETO, 2023))

Dataset name	Link
20Newsgroups	http://qwone.com/$\sim$jason/20Newsgroups/
5AbstractsGroup	https://github.com/qianliu0708/5AbstractsGroup
AG News	http://groups.di.unipi.it/$\sim$gulli/AG_corpus_of_news_articles.html
BBC Corpus	http://mlg.ucd.ie/datasets/bbc.html
BBCSport labeled	http://mlg.ucd.ie/datasets/bbc.html
BlurbGenreCollection	https://www.inf.uni-hamburg.de/en/inst/ab/lt/resources/data/blurb-genre-collection.html
CR	https://www.cs.uic.edu/$\sim$liub/FBS/sentiment-analysis.html
CSTR	http://sites.labic.icmc.usp.br/rsinoara/doc-embeddings/
EURLEX 57K	http://nlp.cs.aueb.gr/software_and_datasets/EURLEX57K/
IMDB	http://ai.stanford.edu/$\sim$amaas/data/sentiment/
JRC-Acquis	https://ec.europa.eu/jrc/en/language-technologies/jrc-acquis
MPQA	http://mpqa.cs.pitt.edu/
MR	https://www.cs.cornell.edu/people/pabo/movie-review-data/
MR (2004)	https://www.cs.cornell.edu/people/pabo/movie-review-data/
MR (2005)	https://www.cs.cornell.edu/people/pabo/movie-review-data/
Ohsumed dataset	http://disi.unitn.it/moschitti/corpora.htm
Ohsumed-400	http://sites.labic.icmc.usp.br/rsinoara/doc-embeddings/
R52	https://github.com/yao8839836/text_gcn/tree/master/data/R52
RCV1-V2	http://www.ai.mit.edu/projects/jmlr/papers/volume5/lewis04a/lyrl2004_rcv1v2_README.htm
Reuters-21578	http://www.daviddlewis.com/resources/testcollections/reuters21578/
Reuters10	https://www.nltk.org/book/ch02.html
Sogou News	http://www.sogou.com/labs/resource/cs.php
SST	http://nlp.stanford.edu/sentiment
TREC	https://cogcomp.seas.upenn.edu/Data/QA/QC/
WIPO-Gama	https://www.wipo.int/classifications/ipc/en/ITsupport/Categorization/dataset/

Table 5 – Other datasets used to train and evaluate text classification approaches ((COSTA; OLIVEIRA; FILETO, 2023))

Dataset name	References
AAPD	Yang et al. 2018 (YANG et al., 2018)
ACL Anthology Network,	Lai et al. 2015 (LAI et al., 2015)
Ads-1m	Prabhu et al. 2014 (PRABHU; VARMA, 2014)
Amazon	Bhatia et al. 2015 (BHATIA et al., 2015)
Amazon Review	Qiao et al. 2018 (QIAO et al., 2018)
Amazon Review Full	Johnson et al. 2015 (JOHNSON; ZHANG, 2015)
Amazon Review Polarity	Johnson et al. 2015 (JOHNSON; ZHANG, 2015)
Amazon-670K	Jiang et al. 2021 (JIANG et al., 2021)
AmazonCat-13k	Jiang et al. 2021 (JIANG et al., 2021)
Bengali corpus	Hossain et al. 2021 (HOSSAIN; HOQUE; SARKER, 2021)
BioASQ	Nam et al. 2016 (NAM; MENCÍA; FüRNKRANZ, 2016)
ChnSentiCorp-Htl-unba-10000	Li et al. 2020 (LI; YE, 2020)
COPD	Liu et al. 2021 (LIU; WANG; REN, 2021)
Czech Corpus	Lenc et al. 2017 (LENC; KRÁL, 2017)
DBLP (academic paper)	Hu et al. 2020 (HU et al., 2020)
DBpedia	Zhang et al. 2015 (ZHANG; ZHAO; LECUN, 2015)
Delicious-Large	Wetzker et al. 2008 (WETZKER; ZIMMERMANN; BAUCKHAGE, 2008)
Emotions	Kumar et al. 2018 (KUMAR et al., 2018)
Enron	Liu et al. 2021 (LIU et al., 2021)
EurLEX	Bhatia et al. 2015 (BHATIA et al., 2015)
EurLEX-4k	Chang et al. 2020 (CHANG et al., 2020)
Fudan Set	Lai et al. 2015 (LAI et al., 2015)
Medical	Liu et al. 2021 (LIU et al., 2021)
MEDLINE citations	Xu et al. 2016 (XU et al., 2016)
Movielens	Liu et al. 2021 (LIU et al., 2021)
Paper theme data	Li et al. 2020 (LI; YE, 2020)
Ren-CECps	Li et al. 2011 (LI; REN, 2011)
Saraswat corpus	Saraswat et al. 2021 (SARASWAT; ABHISHEK; KUMAR, 2021)
SE-product	Sinoara et al. 2019 (SINOARA et al., 2019)
SST-2	Zhao et al. 2018 (ZHAO et al., 2018)
Subjectivity dataset	Zhao et al. 2018 (ZHAO et al., 2018)
Wiki-500K	Chang et al. 2020 (CHANG et al., 2020)
Wiki10	Chang et al. 2020 (CHANG et al., 2020)
Wiki10-31K	Zhang et al. 2021 (ZHANG; YAMANA, 2021)
WikiLSHTC	Bhatia et al. 2015 (BHATIA et al., 2015)
WOS-11967	Kowsari et al. 2017 (KOWSARI et al., 2017)
Yahoo Answer	Johnson et al. 2015 (JOHNSON; ZHANG, 2015)
Yelp Review Full	Johnson et al. 2015 (JOHNSON; ZHANG, 2015)
Yelp Review Polarity	Johnson et al. 2015 (JOHNSON; ZHANG, 2015)

Table 6 – Text classification approaches evaluated with the accuracy score (COSTA; OLIVEIRA; FILETO, 2023)

Work	Dataset																										
	20Newsgroups	5AbstractsGroup	ACL Anthology Network	AG News	Amazon Review	Amazon Review Full	Amazon Review Polarity	Amazon-670K	AmazonCat-13K	BBC corpus	BBCSport labeled	Bengali corpus	ChnSentiCorp-Htl-unba-10000	CR	DBLP	DBpedia	Eurlex-4K	Emotions	Fudan Set	IMDB	MPQA	MR (2004)	MR (2005)	MR	MPQA	Ohsumed dataset
Lai et al. 2015	**96.49**		**49.19**																**95.2**							
Jin et al. 2016	83.1																									
Kumar et al. 2018																		**55**								34.4
Qiao et al. 2018				92.8		**60.9**	**95.3**									98.9										
Liu et al. 2018	86	**87.2**																		90.8				65.1		
Zhao et al. 2018				92.6										85.1										82.3		
Wang et al. 2018				92.45												99.02										
Zang et al. 2019	76.7															85.2										
Pan et al. 2019																92.1										
Liu et al. 2019	84.4																									
Guo et al. 2019														**87.5**										86.6	**93**	
Shi et al. 2019																										
Bounabi et al. 2020											**99.1**															
Hu et al. 2020															**76,42**									**93.56**		
Li et al. 2020													**89.1**													
Aubaid et al. 2020	90.07																									
Meng et al. 2020				86.4	91.6											91.1				86.5						
Kim et al. 2020	86.74																			89.8		**89**	**81**		90.1	
Pittaras et al. 2021	78.4									**97.6**																**43.55**
Gupta et al. 2020	86.19																									
Saraswat et al. 2021																										
Lee et al. 2021				91.83	**92.94**											**99.03**										
Zhang et al. 2021	87.24			**94.72**																**94.06**						
Jiang et al. 2021								**49.1**	**96.77**								**87.63**									
Hossain et al. 2021												**96.85**														

Table 6 (continuation): Text classification approaches evaluated with the accuracy score (COSTA; OLIVEIRA; FILETO, 2023)

Work	Dataset														
	Paper theme data	R52	Reuters-21578	Reuters10	Saraswat corpus	Sogou News	SST	SST-2	Subjectivity dataset	TREC	Wiki-500K	Wiki10-31K	Yahoo Answer	Yelp Review Full	Yelp Review Polarity
Lai et al. 2015							47.21								
Kumar et al. 2018															
Qiao et al. 2018						**97.6**							73.7	**64.9**	**96.4**
Liu et al. 2018							**78.8**								
Zhao et al. 2018								**86.8**	93.8	92.8					
Wang et al. 2018													**77.42**	64.09	95.31
Zang et al. 2019															
Pan et al. 2019															
Liu et al. 2019			90.2												
Guo et al. 2019									**95.6**	91.9					
Shi et al. 2019						86.53									
Bounabi et al. 2020															
Hu et al. 2020		**97.23**													
Li et al. 2020	**97.7**					97.5									
Aubaid et al. 2020			90.72												
Meng et al. 2020															
Kim et al. 2020				**87.52**						**94.8**					
Pittaras et al. 2021			74.9												
Gupta et al. 2020															
Saraswat et al. 2021					**93**										
Lee et al. 2021													74.13		
Zhang et al. 2021															
Jiang et al. 2021											**77.78**	**89.45**			
Hossain et al. 2021															

Table 7 – Text classification approaches evaluated with other metrics instead of accuracy (COSTA; OLIVEIRA; FILETO, 2023)

Metric	Work	Dataset																							
		20Newsgroups	5AbstractsGroup	Ads-1m	AAPD	Amazon	AmazonCat-13k	BBC Corpus	BioASQ	BlurbGenreCollection	COPD	CSTR	Czech Corpus	Delicious-Large	Emotions	Enron	EurLEX	EurLEX-4k	EURLEX 57K	JRC-Acquis	Medical	MEDLINE citations	Movielens	MR
Macro F1	Lai et al. 2015	**96.49**																						
	Xu et al. 2016	95																				**50**		
	Jin et al. 2016	82.7																						
	Kumar et al. 2018														**64.9**									
	Sinoara et al. 2019							98.01				**83.44**												
	Pittaras et al. 2021	79						**97.6**																
	Liu et al. 2021				**57.1**						**94.4**													
Micro F1	Kumar et al. 2018														**68.2**									
	Chalkidis et al. 2019																		**69.8**					
	Sinoara et al. 2019							97.98				**82.63**												
	Liu et al. 2019	**83.9**																						
	Liu et al. 2021				**72.35**											**73.91**					**92.22**		**83**	
	Liu et al. 2021				72.1						**93.5**													
F1	Lenc et al. 2017												**84.19**											
	Liu et al. 2018	85	**87.1**																					
	Aly et al. 2019									**74.37**														
	Pappas et al. 2019								**79.85**															
	Liu et al. 2019	**90**																						
	Hu et al. 2020																							**93.57**
	Aubaid et al. 2020	70.98																						
	Cai et al. 2020				**74.4**																			
	Gupta et al. 2020	86.16																						
	Moreo et al. 2021	70.7																		**39.7**				
Precision	Lenc et al. 2017												**87.67**											
	Liu et al. 2018	85.5	**86.2**																					
	Aly et al. 2019									**77.21**														
	Hu et al. 2020																							**93.59**
	Aubaid et al. 2020	76																						
	Cai et al. 2020				**76.8**																			
	Gupta et al. 2020	**86.2**																						
	Chang et al. 2020						**96.7**											**87.22**						
	Bhatia et al. 2015			**21.84**		**35.05**								**47.03**			**80.17**							
Recall	Lenc et al. 2017												**83.55**											
	Liu et al. 2018	85	**87.1**																					
	Aly et al. 2019									**71.73**														
	Hu et al. 2020																							**93.56**
	Aubaid et al. 2020	66.64																						
	Cai et al. 2020				**72.2**																			
	Gupta et al. 2020	**86.18**																						

Table 7 (continuation): Text classification approaches evaluated with other metrics instead of accuracy (COSTA; OLIVEIRA; FILETO, 2023)

Metric	Work	Dataset										
		Ohsumed-400	Ohsumed dataset	RCV1-V2	Ren-CECps	Reuters-21578	SE-product	Wiki10(30K labels)	Wiki-500k	WikiLSHTC	WIPO-Gama	WOS-11967
Macro F1	Lai et al. 2015											
	Xu et al. 2016											
	Jin et al. 2016					**88.6**						
	Kumar et al. 2018		27.9									
	Sinoara et al. 2019	**38.21**					**95.3**					
	Pittaras et al. 2021		**37.3**			37.8						
	Liu et al. 2021				**58.5**	67.5						
Micro F1	Kumar et al. 2018		46.3									
	Chalkidis et al. 2019											
	Sinoara et al. 2019	**37.96**					**99.26**					
	Liu et al. 2019					90.3						
	Liu et al. 2021		**77.88**									
	Liu et al. 2021				**69.2**	**90.8**						
F1	Lenc et al. 2017					87.59						
	Liu et al. 2018											
	Aly et al. 2019											**81.69**
	Pappas et al. 2019											
	Liu et al. 2019					**93**						
	Hu et al. 2020											
	Aubaid et al. 2020					76.75						
	Cai et al. 2020			**89.9**								
	Gupta et al. 2020					82.71						
	Moreo et al. 2021		**73.1**	69.5		65.2					**57.1**	
Precision	Lenc et al. 2017					91.03						
	Liu et al. 2018											
	Aly et al. 2019											**82.75**
	Hu et al. 2020											
	Aubaid et al. 2020					79						
	Cai et al. 2020			**90.6**								
	Gupta et al. 2020					**95.06**						
	Chang et al. 2020							**88.51**	**77.28**			
	Bhatia et al. 2015							85.54		**55.57**		
Recall	Lenc et al. 2017					86.14						
	Liu et al. 2018											
	Aly et al. 2019											80.67
	Hu et al. 2020					**93.56**						
	Aubaid et al. 2020					75.9						
	Cai et al. 2020			**89.2**								
	Gupta et al. 2020											

attention mechanism (BAHDANAU; CHO; BENGIO, 2015), Transformers (VASWANI et al., 2017), BERT (DEVLIN et al., 2019), and XLNet (YANG et al., 2019), among others. However, despite these progresses, there are still challenges to be addressed.

For the best of our knowledge, current trends and research opportunities for text classification research have not been fully described yet. Thus, based on our experience and what we have identified in our bibliographical review, we point out the following themes as promising directions:

Cost-effective text classification models: Usually, text classification is formulated as a supervised learning problem, where a labeled dataset is used to train a classifier. However, training a supervised neural network model can be resource-intensive, requiring expensive hardware with sufficient memory and GPU capabilities. Additionally, obtaining extensive datasets for training purposes can be challenging, as it often involves a substantial amount of human labor to create and annotate the data. To meet the computation and storage restrictions of several users and applications, the models have to be compressed. One way it can be done is by building learner models using knowledge distillation or modeling compression techniques (WANG et al., 2020). Moreover, the fine-tuning of the existing models for the necessities of an application may alleviate the expensive hardware and dataset size requirements while providing significant improvements.

Effective multi-label classification: Traditional multi-label classification methods do not deal adequately with the increasing needs of contemporary big and complex data structures. As a result, there is a critical need for new multi-label learning paradigms, and new trends are emerging (LIU et al., 2021). Extreme multi-label classification (XMLC) becomes a developing new line of research that focuses on multi-label problems with a vast number of labels. The existing multi-label classification techniques do not address the XMLC problem due to the prohibitive computational cost. Analyzing all the positive labels to text documents poses a challenge in XMLC. An issue in multi-label classification is modeling the interdependencies between labels and features. Existing methods attempt to model the correlations between labels and features. Nevertheless, the statistical properties of these multi-label dependency modelings are less explored, and theoretical analysis is a necessary future research topic.

Use of KGs and Knowledge Embedding: We envision that knowledge bases such as KGs may contribute to future text classification approaches. Most text classification approaches that exploit embeddings employ only word embeddings, as discussed in Section 3.4. Only a few works use KGs to improve the word embeddings, like (SINOARA et al., 2019) and (ZHANG; LERTVITTAYAKUMJORN; GUO, 2019). On the other hand, KGs contain millions of facts describing with precise semantics entities mentioned in the texts to be classified. They can be exploited more efficiently through KG embeddings to improve text classification. Word embeddings and knowledge embeddings are usually trained in independent ways by using different techniques. Consequently, their respective embeddings are in different vector spaces, hindering their joint use. Nevertheless, these incompatibilities can be overcome by several techniques that combine embedding techniques.

4 THE OPHELIA APPROACH

Our research originated from a significant gap identified in the literature — a paucity of investigations into the intricate semantic relationships among words in textual contexts. Importantly, this research endeavor underscores the vital importance of delving into not only word embeddings but also knowledge embeddings, both jointly trained within the same vector space. This dual emphasis reflects our commitment to understanding the nuanced complexities of language and domain-specific knowledge representation, searching to yield a richer understanding of semantic relationships in a text.

Our proposal, OPHELIA (knOwledge GraPH-augmented tExt cLassIfication Approach), uses word embeddings and knowledge embeddings jointly trained in a neural network for text classification. It was developed considering the key aspects discussed in subsection 3.2.1. OPHELIA employs jointly trained knowledge and word embeddings as features to feed a neural network model that classifies texts according to their categories. Therefore, OPHELIA aims to tackle the interdependent approach for the text classification task by seamlessly combining several inputs (words and entities) through jointly trained word and knowledge embeddings. This proposal is derived from the research questions and objectives stated in Section 1.3 and the analysis of recent related work discussed in Section 3.4.

Figure 13 provides an overview of the OPHELIA architecture. It can be divided into three major modules: *Embedding Generation*, *Semantic Enrichment*, and *Text Classification*. The *Text Classification* module employs the neural network using text features in the form of joint embeddings. The following sections detail each one of these modules. The decomposition of the approaches in three modules greatly enhances our ability to grasp the intricate interplay among the various elements of the system. Moreover, this modularization empowers us to work on each component independently, streamlining the process of system maintenance and updates. In summary, the partitioning of a project into modules represents a fundamental programming best practice, enhancing organization, simplifying maintenance, bolstering adaptability for future developments, and bringing a better understanding of the process proposed.

4.1 EMBEDDING GENERATION MODULE

The embedding generation in our current implementation is done by using fastText (BOJANOWSKI et al., 2017; JOULIN et al., 2017a; JOULIN et al., 2017b), which is available in Github[1]. In Section 2.4.3, we delve into the key factors guiding our choice of FastText as the embedding framework. FastText's implementation grants the unique capability for each entity or concept to be linked to one or more textual property values. In the conventional FastText approach, these textual properties are leveraged by considering character n-grams. This distinctive feature paves the way for the concurrent training of word embeddings and knowl-

[1] https://github.com/facebookresearch/fastText

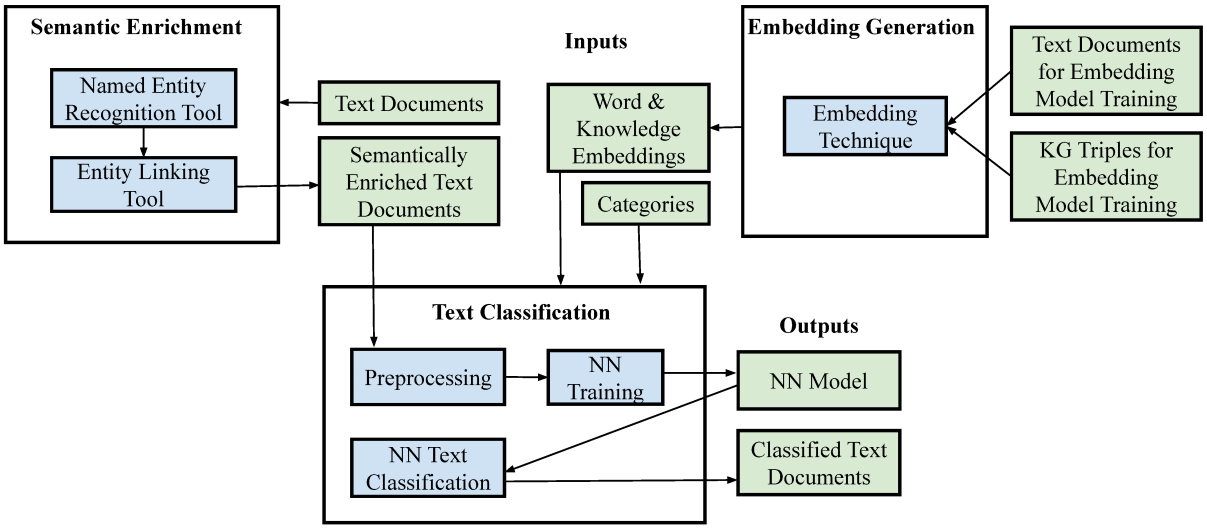

Figure 13 – Overview of OPHELIA main modules and text classification process.

edge embeddings through the FastText framework. It's worth noting that FastText consistently delivers competitive outcomes in both word embedding and knowledge embedding tasks.

The fastText (JOULIN et al., 2017b) model uses a linear model with ranking constraints. The weight matrix V is utilized as a look-up table for the discrete tokens, and the weight matrix W is used for the classifier. The representations of the discrete tokens are averaged into a Bag-of-Words (BoW) feature representation. The average representation of each word is then fed to the linear classifier. Using a softmax function f to compute the probability distribution over the classes, and considering N input sets for a discrete token (e.g., sentences), leads to the minimization of the following function:

$$-1/N \sum_{n-1}^{N} \mathrm{y}_n log(f(WV\mathrm{x}_n)), \tag{4.1}$$

where x_n is the normalized bag of features of the n-th input set and y_n its label. While BoW models are memory inefficient, their memory footprint can be significantly reduced. The model is trained asynchronously on multiple CPUs with a Stochastic Gradient Descent (SGD) optimizer and a linearly decaying learning rate.

As presented in Figure 13, KG triples and entity abstracts are used as inputs of fastText to jointly train knowledge embeddings and word embeddings in the same vector space. The knowledge graph (KG) chosen for this work is DBpedia because of its open-domain nature, which allows us to evaluate our approach using different open-domain corpora. We use only the high-quality facts provided by DBpedia[2] to avoid noise in our embedding representation. As textual entity descriptions, we use the long version of the DBpedia entity abstracts[3], which contains the introductory text of each Wikipedia page and provides a summary of the respective entity.

[2] http://wiki.dbpedia.org/services-resources/documentation/\\datasets\#MappingbasedObjects
[3] http://wiki.dbpedia.org/services-resources/documentation/\\datasets\#LongAbstracts

We have combined infobox data triples and long abstracts of entities in a single training file. This allows fastText to jointly produce knowledge and word embeddings in the same vector space. The parameters for the fastText model training are detailed and discussed in Chapter 5.

4.2 SEMANTIC ENRICHMENT MODULE

The Semantic Enrichment Module enriches text documents by performing Entity Linking (EL). It links named entity mentions found in text documents with their respective entity descriptors in KGs like DBpedia. Prior to EL, it is necessary to do Named Entity Recognition (NER), as illustrated in Figure 13. However, several EL tools also incorporate NER methods. Thus, we just assume that the NER task will always be performed, either by a specific NER tool or by a tool that does NER and EL (also called end-to-end EL).

In this work, we have used Babelfy (MORO; RAGANATO; NAVIGLI, 2014) to semantically enrich the text documents by linking named entities in the text to their corresponding entities in DBpedia. Babelfy performs two NLP tasks, namely end-to-end EL and Word Sense Disambiguation (WSD). WSD aims to disambiguate the meaning of ordinary words, such as nouns and verbs, and link them to their correct sense in a knowledge base. Babelfy leverages the BabelNet knowledge base to perform these tasks.

BabelNet is an extensive multilingual encyclopedic dictionary and ontology which covers 50 languages. Based on the automatic integration of lexicographic and encyclopedic knowledge extracted from multiple sources (WordNet, Wikipedia, Open Multilingual WordNet, OmegaWiki, Wiktionary, and WikiData), BabelNet offers a vast network of words and entities along with an extensive multilingual lexical coverage. The last version of BabelNet is available upon request[4]. Another reason we use Babelfy is that it is available as a Web service that can be used to enrich one thousand documents per day for free. This allows us to save computational resources.

Semantic enrichment can be done either online or offline. We consider a task to be online when it is integrated into our text classification system, and there is no need for a user to call it manually and take the results into our system. Offline tasks, on the other hand, require text documents to be semantically enriched before the execution of our approach. An example of an offline task in our approach is the *Embedding Generation*, which has to be done beforehand.

The semantic enrichment of the textual documents used as the training dataset for the neural network in our experiments is also done offline so that we can submit batches of text documents at once and, consequently, save time and computational resources.

4.3 TEXT CLASSIFICATION MODULE

The first task of the Text Classification Module is to preprocess the semantically enriched texts. Typically, for NLP tasks, preprocessing is the initial step to be performed on a

[4] https://babelnet.org/

text document. However, as shown in Figure 13, we perform the preprocessing after the semantic enrichment of the text document because the Named Entity Recognition (NER) and Entity Linking (EL) tools already include preprocessing steps in their implementations. Thus, if we preprocessed the text documents before their semantic enrichment, it could have a negative impact on the results of NER and EL tools.

The main goal of our preprocessing task is to fix words to match their respective embedding representations. To achieve this, we remove capitalization, special characters, apostrophes, and stop words. Capitalization and special characters are removed by using Python built-in functions, like *to_lower()* and *isalnum()*. Stop words are removed using the list provided by libraries like nltk[5]. Lastly, we remove possessive apostrophes when they appear as "'s". All these preprocessing steps are represented by line 3 in Algorithm 1. Although simple, it is essential to highlight that excessive preprocessing could lead our approach to use incorrect embedded representations. For example, if we transform the verb "died", conjugated in the past, into its infinite "die", the neural network may misinterpret the sentence due to the different embedding representations. The embedded representation of "died" should be more closely related to entities that already have passed away than to the embedded representation of "die".

In the embedded representations of the semantically enriched texts, entity mentions are represented by their respective entity embeddings, while the other words are represented as word embeddings. It's important to note that the embeddings are trained jointly in the same vector space using fastText. They are then used as features to train the neural classifier and classify documents.

The neural network is the main component of the Text Classification Module. Figure 14 presents our two alternative neural network models for text document classification exploiting joint embeddings of words and knowledge.

Model (a) is the most traditional for NLP tasks: a Feed Forward neural network, where information always goes forward, as presented in 2.5.1. Model (b) uses a capsule network as proposed by (KIM et al., 2020), presented in 2.5.2. According to them, while Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) have shown good results in existing works, they fail to capture hierarchical relations between local features and spatial relationships effectively. Therefore, they propose the use of capsule networks to overcome these limitations. Although capsule networks have been mainly used in image processing, the authors argue that they can be successfully applied to text classification, as they consider the spatial relationship between entities. In contrast to their approach, which only considers word embeddings, our approach can take full advantage of spatial relationships by exploring the knowledge embeddings of the entities mentioned in a text. The spatial relationships expressed by entities in a text document can provide useful patterns for improving text document classification.

The first architecture in Model (b) is a gate layer. The gate selects which input features will be considered in their respective instance and preserve spatial information. The Convolu-

[5] https://www.nltk.org/

tional Capsule layer considers both hierarchical relationships (convolutional) and spatial relationships (capsule). For the next layer, Text Capsule, (KIM et al., 2020) proposes a dynamic and static routing. In this work, we consider only static routing, as it presented the best results in (KIM et al., 2020). Finally, differently from (KIM et al., 2020) that uses L2 form to translate the output of the Text Capsule layer into categories, we employ a Feed-Forward architecture. We adopt this strategy because this architecture has been successfully used as the last layer in several NLP tasks and image-processing approaches that employ capsule networks (SABOUR; FROSST; HINTON, 2017).

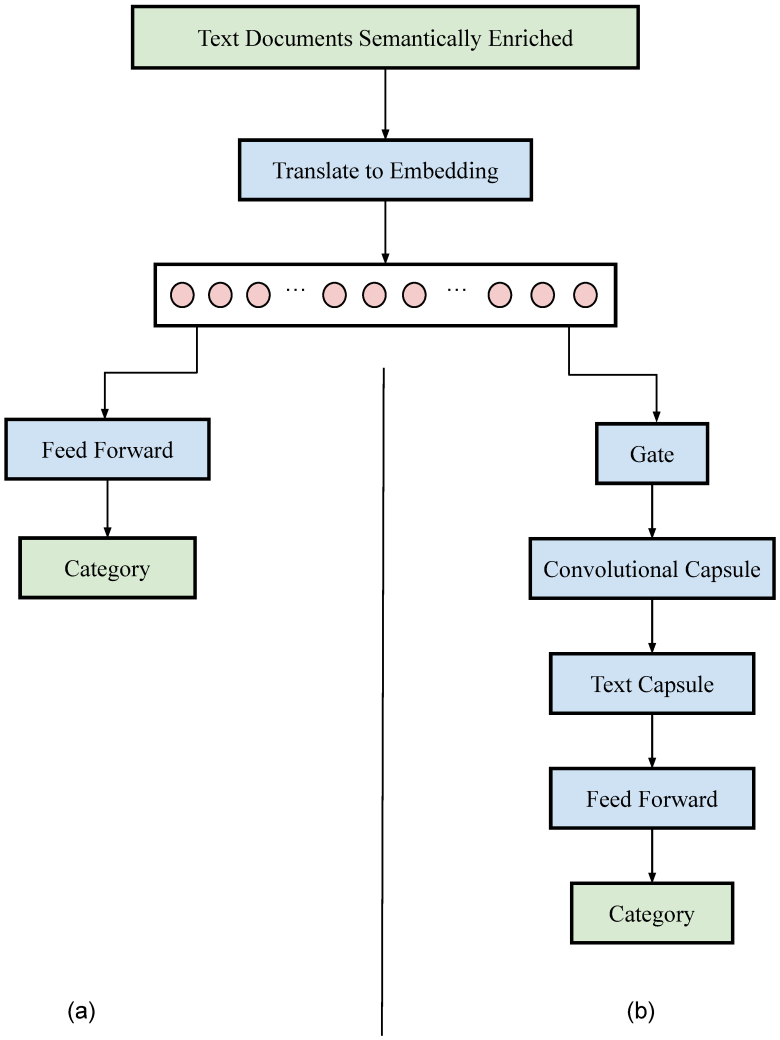

Figure 14 – Text classification exploiting joint embeddings of words and knowledge in alternative neural network models: (a) Feed Forward network, the most traditional model for NLP, and (b) the CapsNet-based model (KIM et al., 2020).

Algorithm 1 describes the classification of semantically enriched documents employing the previously described preprocessing and our trained FFNN and Capsnet model.

Algorithm 1 Text classification process

Input: **D** // List of semantically annotated documents, where $d_i \in D$
E // Set of word and knowledge embedding, where $e_j \in E$
C // Set of predefined classes, where $c_k \in C$

Output: $< d_i, c_k >$ # correct class c_k to classify a text document d_i

1: $R = \emptyset$ # Set that will contain the the pair $< d_i, c_k >$
2: **for** $d_i \in D$ **do**
3: d_i = preprocessing(d_i)
4: c_k = classification(NN_Model(), d_i, C)
5: append(R, $< d_i, c_k >$)
6: **end for**

5 EXPERIMENTAL EVALUATION

This chapter reports the experiments performed to evaluate OPHELIA in the text classification task and discusses the results. Section 5.1 details the experimental setup, including the tools, evaluation metrics, datasets, parameters, and hardware used. Section 5.2 presents the results of the neural network training. These results are used to choose which parameter better fits the neural network model employed in OPHELIA. Section 5.3 presents the OPHELIA results for the neural network architectures FFNN and CapsNet. Lastly, in Section 5.4, we discuss the results presented.

5.1 EXPERIMENTAL SETUP

The experimental setup includes (*i*) the metric used for the evaluation; (*ii*) the parameters for the embedding generation; (*iii*) the parameters for the training of the NN models; (*iv*) the training, validation, and test sets used in all the experiment processes; and (*v*) the configuration of the hardware running the experiments.

We compare OPHELIA results with those of state-of-the-art text classification approaches from the literature by using accuracy and F1 score as the comparison metrics. Some approaches use two versions of the F1 score: micro and macro. The micro F1 score calculation considers all true positives, false positives, and false negatives from all documents together, while the macro F1 score is the average of the F1 scores calculated for each document.

Our neural network model using FFNN obtains the best results using 50-dimensional embeddings. This way, it is possible to train our model in machines with limited hardware. Meanwhile, the DNN model using CapsNet obtained the best results using 100-dimensional embeddings. For the training of our model, we use Adam loss optimization (KINGMA; BA, 2014) with a learning rate of 0.001 and a batch size of 32.

For the embedding generation, we employ fastText with 500 epochs and a context window size of 50. The remaining fastText parameters (e.g., learning rate, number of negative examples, loss function) are set to the default values presented in the fastText GitHub repository[1]. The embedding training dataset that we have used is the one described in Section 4.1.

The values for the learning rate, batch size, and threshold were obtained in preliminary experiments, in which we increased each one of these values until the result quality dropped (for the learning rate and threshold) or the hardware could not support it anymore (for the batch size). The DNN framework used to develop our proposal was pyTorch[2]. PyTorch allows for improving the embedding representation during the model's training by updating its weight matrix. However, we had to turn off this functionality due to the memory limitation of the GPU. The use of such functionality could slightly improve the results of OPHELIA.

[1] https://github.com/facebookresearch/fastText
[2] https://pytorch.org/

To facilitate performance comparison in this study, we utilized three datasets: BBC News, AG News, and Reuters-21578, all of which focus on news articles. Each document within these datasets was semantically enriched using the Babelfy tool (MORO; RAGANATO; NAVIGLI, 2014).

We train a neural network model per dataset. It is necessary because each dataset contains a different number of classes, and their respective text documents present distinct writing styles. We highlight that, in case the dataset does not separate a validation portion, we randomly split the training dataset into 80% of training samples and 20% of validation samples.

BBC News: The BBC News dataset is a collection of 2225 documents published between 2004 and 2005. It consists of news articles collected from the BBC website, representing real-world news content. The dataset is primarily written in English, making it suitable for English text processing and classification tasks. Each article is associated with a corresponding category label, covering five categories: business, entertainment, politics, sport, and tech. The number of documents in each category is (i) Business: 510 (ii) Entertainment: 386 (iii) Politics: 417 (iv) Sport: 511 (v) Tech: 401.

The dataset is preprocessed and tokenized, with stemming (Porter algorithm), stop-word removal (using a stop-word list), and low-term frequency filtering (count < 3) already applied. This preprocessing allows for easier processing and analysis of the text data. The articles vary in length, ranging from a few sentences to several paragraphs, enabling the testing of model robustness with both short and long documents.

The BBC News dataset is well-known for its balanced distribution of articles across categories, ensuring a comparable number of samples in each category. This balance helps avoid class imbalance issues during model training and makes it a valuable resource for training and evaluating text classification models. The dataset is commonly split into training and testing subsets, with approximately 80% used for training and the remaining 20% for testing and evaluation.

This dataset is provided for non-commercial and research purposes, serving as a benchmark for machine learning research in news classification and analysis. Its availability and authenticity make it relevant for practical applications. Researchers and practitioners can utilize this dataset to develop and compare different machine learning algorithms and techniques, showcasing improvements in accuracy and performance metrics.

AG News: The AG News dataset is a collection of news articles obtained from the AG's corpus, which contains over 1 million news articles gathered from more than 2000 sources. The dataset was compiled by ComeToMyHead, an academic news search engine active since July 2004 (ZHANG; ZHAO; LECUN, 2015).

The dataset comprises articles from four major categories: world, sports, business, and science/technology. These categories were chosen as the four largest classes from the original corpus. In total, the dataset contains 120.000 samples, with each class having

30.000 training samples. Each sample consists of a headline and a brief description. The dataset is labeled, ensuring each sample is associated with the appropriate category label. Furthermore, the distribution of samples across categories is balanced, providing an equal number of samples for each category.

Reuters-21578: The Reuters-21578 was released in 1987 and contains news articles from the Reuters news agency. The dataset is provided in a standardized SGML format, where each article is encoded within an SGML document structure. Preprocessing steps, such as removing HTML tags, tokenization, stemming, and stop-word removal, it is necessary before utilizing the dataset for text classification tasks. The original Reuters-21578 dataset is publicly available for research purposes. However, due to its age, researchers have created variations and adaptations over time, incorporating additional preprocessing or modifications to cater to specific research needs.

The articles in the dataset vary in length, ranging from a few sentences to several paragraphs. This diversity in document length presents challenges and opportunities for text classification algorithms to handle documents of varying sizes. The structure of this format is described in (LEWIS et al., 1987).

The dataset comprises a collection of 21.578 news articles that are classified into different predefined categories or topics. Each article in the dataset includes various fields, such as title, dateline, body text, topics, and metadata. The dataset covers a wide range of topics, including business, finance, economics, politics, sports, health, and more. These topics represent the subjects or categories to which the news articles belong. In this work, we simplified the original dataset, which had a hierarchical structure with multiple levels of categories, to a single-label classification problem for text classification tasks. We focused solely on the first topic of each document as the correct class.

We have used blades of the SDumont supercomputer[3] for embedding generation and neural network training. The embedding generation was done on blades having just CPUs, while the training runs on blades also having GPU. The first blades have 2 CPU Intel(R) Xeon(R) E5-2695v2 @ 2.4 GHz with 12 cores and 64 GB DDR3 1866MHz RAM memory. The other blades have 2 CPU Intel(R) Xeon(R) E5-2695v2 @ 2.4 GHz with 12 cores, 64 GB DDR3 1866MHz RAM memory, 2 GPU Nvidia Nvidia K40, 3584 Cuda cores, and 16GB of memory. However, we highlight that we were able to run the experiments for evaluating the trained NN performance on text classification in a Dell notebook with 1 CPU Intel(R) Core i3 and 20 GB DD3 1866MHz RAM memory.

[3] https://sdumont.lncc.br/

5.2 NEURAL NETWORK EVALUATION

The parameter variation in the training of a NN can influence its result quality, i.e., how well a NN performs the task it was trained for. Therefore, in this section, we present the evaluation results of different NN training configurations.

Prior to the DNN evaluation, we highlight we employ 50-dimensional embeddings to evaluate our FFNN architecture and 100-dimensional embeddings to evaluate our CapsNet architecture. The embeddings' dimensions were obtained in empirical experiments. For the DNN, we train several models by varying the number of hidden dimensions and the number of hidden layers. For hidden dimensions, we start at 50 and increase this number by 50 until reaching 200. For the number of hidden layers, we use the values 4, 10, and 20.

Figure 15 and Figure 16 show the values of the training loss and valid loss during the training of the FFNN and CapsNet, respectively. It is possible to notice that the training stabilizes before epoch 20 and epoch 25. After the stabilization, the valid loss is always slightly above the training loss. Therefore, the high number of epochs does not cause overfitting or underfitting. The models training in the other datasets presents equivalent graphical results.

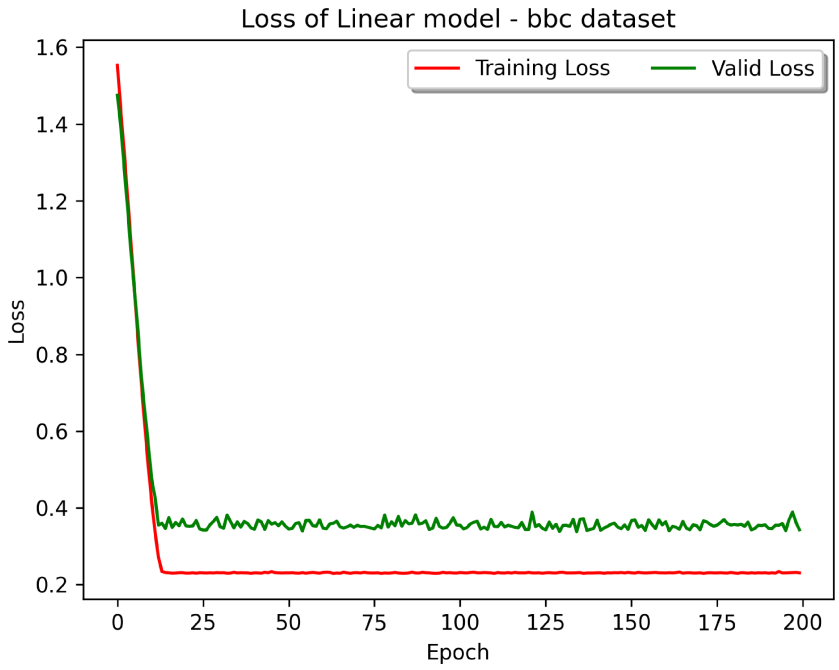

Figure 15 – Training and valid loss during training of an FFNN. The training lasted 200 epochs. The red line represents the training loss. The green line represents the valid loss. The y-axis is the loss value. The x-axis is the number of epochs.

5.3 OPHELIA EVALUATION

In this study, we present an analysis of different text classification models and their performance on three distinct datasets: BBC News, AG News, and Reuters-21578. Through an

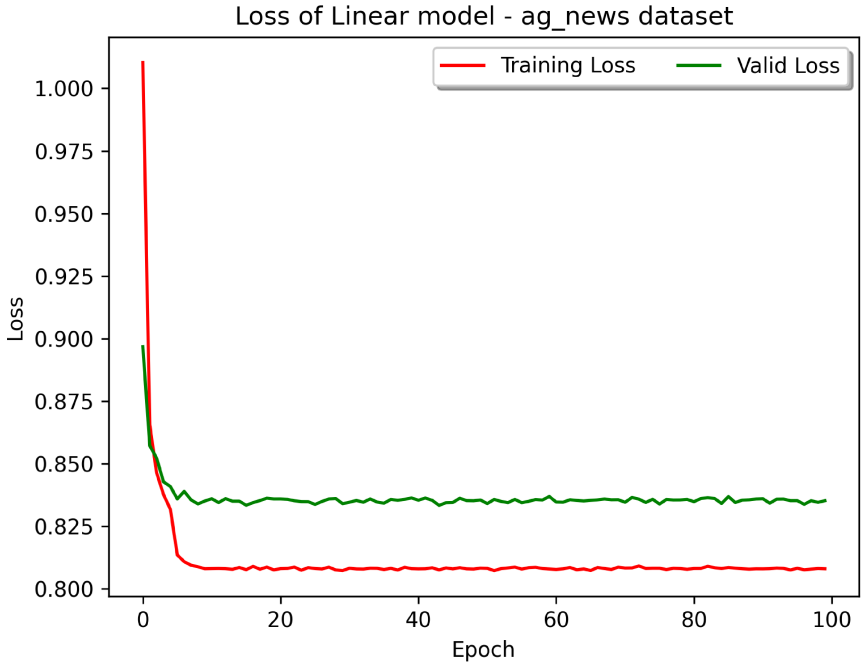

Figure 16 – Training and valid loss during training of a CapsNet. The training lasted 100 epochs. The red line represents the training loss. The green line represents the valid loss. The y-axis is the loss value. The x-axis is the number of epochs.

evaluation, we aim to compare it with existing approaches from the literature. Table 8 presents the accuracy, micro and macro F1 scores (lines *F1@micro* and *F1@macro*, respectively) of our proposal and of state-of-the-art approaches.

In our experiment, aimed at generating the results featured in Table 8, we designed and optimized both a Feedforward Neural Network (FFNN) and a Capsule Network (CapsNet) configuration. In the FFNN setup, we prioritized the balance between computational efficiency and model convergence, opting for a batch size of 32. An embedding size of 200 was chosen to empower the model to discern intricate textual features, thereby enabling the acquisition of rich and informative representations. Over the course of 50 epochs, our network iteratively honed its weights and biases, with a judiciously selected learning rate of 0.001 to ensure efficient learning and stability. Additionally, our experiment also incorporated a Capsule Network (CapsNet), where we fine-tuned an array of hyperparameters to optimize its performance. The CapsNet featured a batch size of 32 and an embedding size of 100, undergoing training for 100 epochs with a learning rate of 0.001. The architecture was composed of 20 layers, each housing 50 hidden cells, and a dropout rate of 0.5 was thoughtfully applied to prevent overfitting. We employed the "caps" model type, incorporating 10 capsules within the capsule layer, each with a dimensionality of 16. A regularization strategy was enforced with a dropout rate of 0.28. These configurations were instrumental in achieving the best results in our experiment, showcasing its effectiveness in text classification.

In addition to accuracy and F1 scores, we use a confusion matrix to evaluate the performance of our proposal in regard to each class of the datasets. The confusion matrix is a useful

tool for evaluating the performance of a classification model by presenting the number of true positives and false positives for each class.

By analyzing the confusion matrices of different datasets, we can gain insights into the classification performance of the models trained on these datasets. In this work, we compare the confusion matrices of the BBC News dataset, AG News dataset, and Reuters-21578 dataset. This evaluation technique allows us to assess the model's ability to correctly classify instances from different classes and identify any potential areas of improvement.

The BBC News dataset exhibits the best performance, as indicated in Table 8, outperforming the existing approaches. The high accuracy can be attributed to several factors. The dataset's preprocessing, which includes tokenization, stemming, and stop-word removal, further enhances the quality of the data and aids in accurate classification. The dataset's balanced distribution of samples across categories ensures that each class has a comparable number of instances, minimizing the risk of class imbalance issues during training. And, observing empirically the texts of each class, it is possible to notice that they are very distinct from each other.

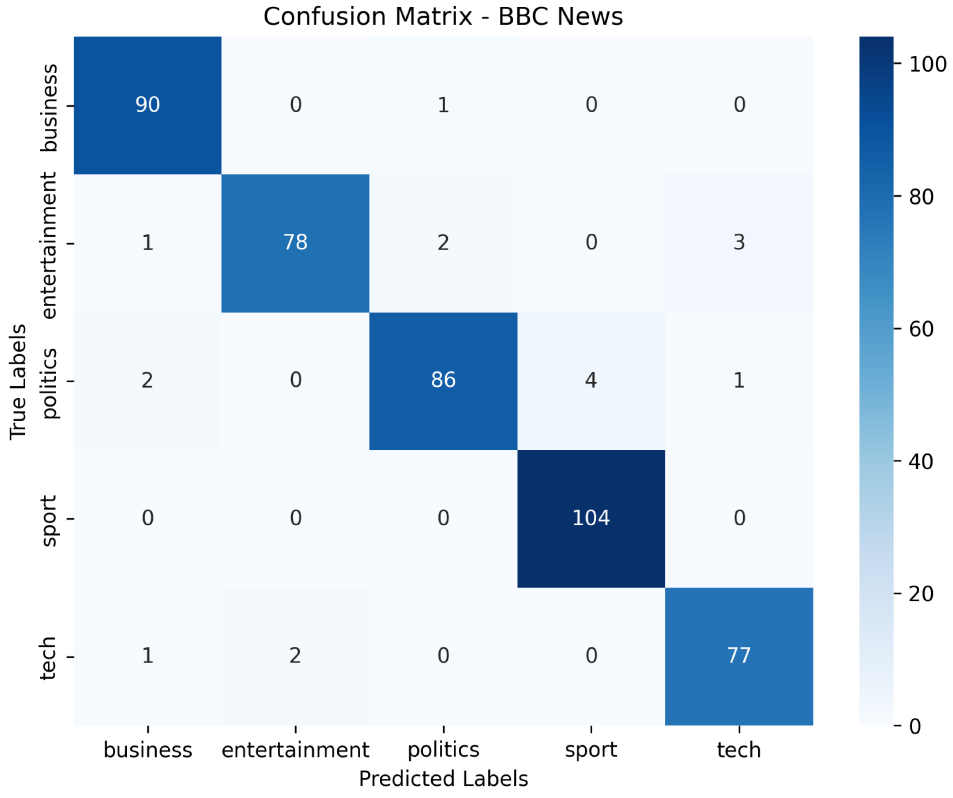

Figure 17 – Confusion matrix obtained on BBC News dataset.

The AG News dataset performs reasonably well, staying competitive with the existing approaches, although not as well as the BBC News dataset, as indicated by Table 8. As the BBC News dataset, the AG News dataset's balanced distribution of samples and its size contribute to a fair and reliable evaluation of text classification models. However, as it is possible to notice in its confusion matrix (Figure 18), OPHELIA classifies with fewer precision texts from the

classes business and science/technology.

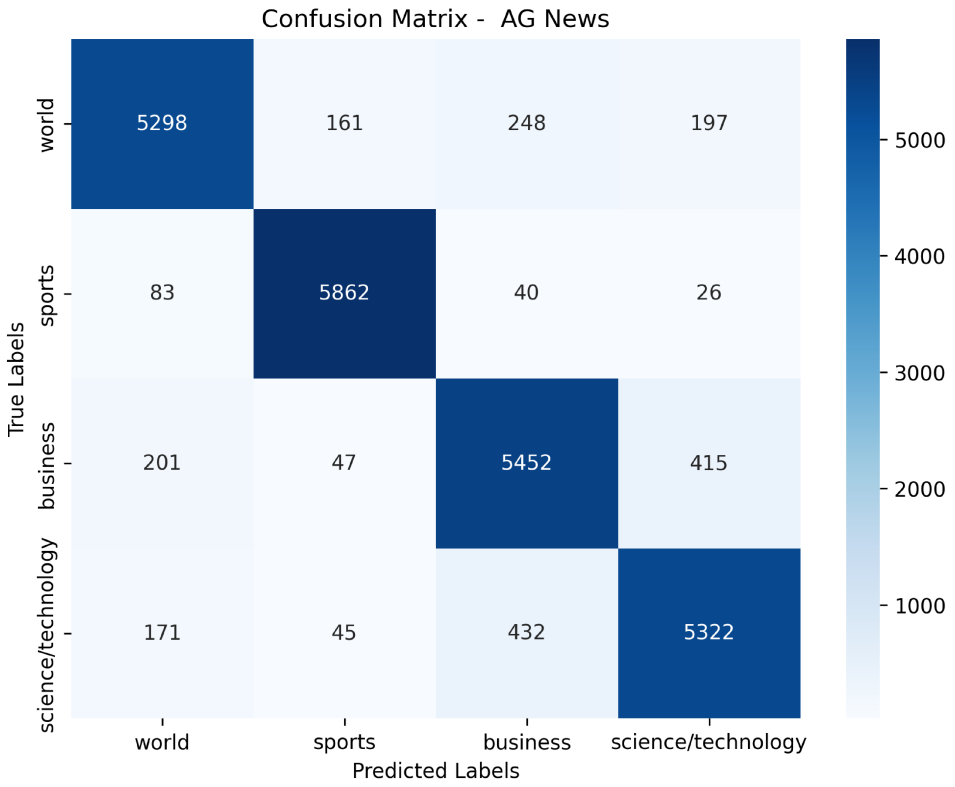

Figure 18 – Confusion matrix obtained on AG News dataset.

The Reuters-21578 dataset exhibits the lowest performance among the three datasets, although it also stays competitive with existing approaches, as evident from Table 8. This dataset, which consists of news articles from Reuters, has certain characteristics that pose challenges for classification models. For example, the dataset's classes have imbalanced distributions (as highlighted in Figure 19), making it harder to achieve accurate predictions for certain categories. Moreover, the dataset's preprocessing and quality vary, affecting the model's performance. For example, the texts frequently have acronyms with no equivalent in our embeddings. It is important to note that the Reuters-21578 dataset is a widely used benchmark dataset, but its performance in our evaluation suggests that it requires additional preprocessing or more sophisticated modeling techniques to achieve better results.

5.4 DISCUSSION

This section discusses the results of OPHELIA besides the scores presented in Table 8. Moreover, we also discuss the challenges to propose and develop a new approach for the text classification task.

The text classification performance was evaluated with the metrics accuracy and F1 score because they are the most used in text classification approaches that we have considered as baselines (PITTARAS et al., 2021; LENC; KRÁL, 2017; LEE; LEE; YU, 2021; ZHANG; YAMANA, 2021). Table 8 presents the accuracy, micro and macro F1 scores of our proposal,

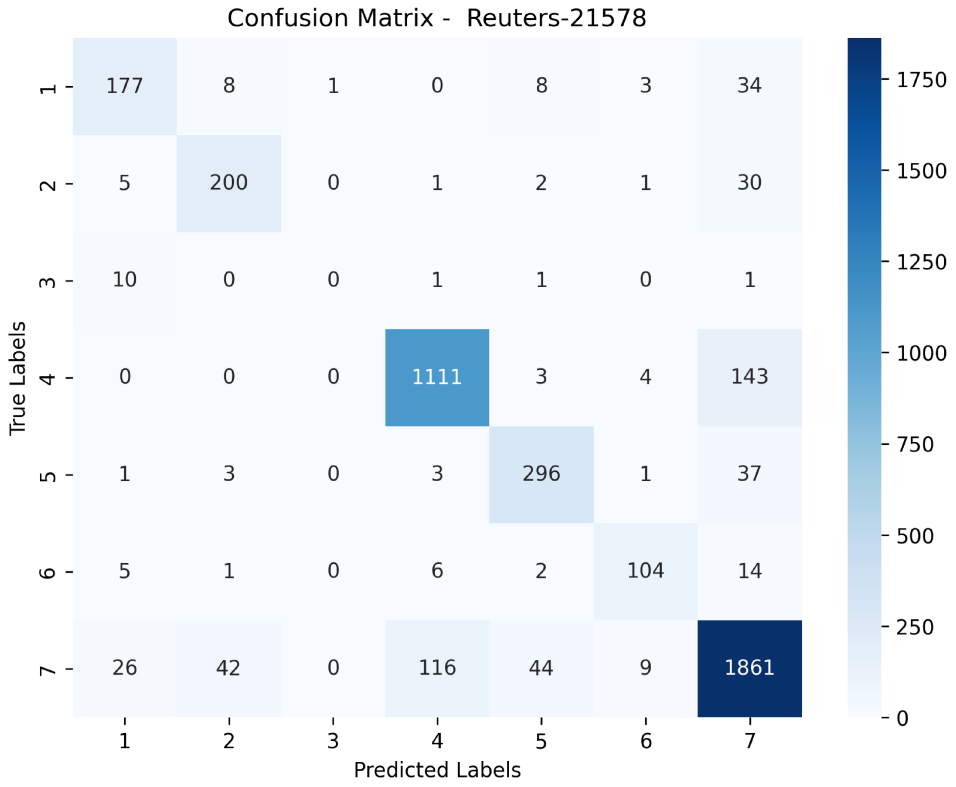

Figure 19 – Confusion matrix obtained on Reuters-21578 dataset.

and state-of-the-art approaches used as baselines. We experiment OPHELIA with different embedding dimensions. The best results with FFNN were possible using embedding of 50 dimensions. Meanwhile, the best results with CapsNet were possible using embedding of 100 dimensions.

The best measures for each dataset are highlighted in bold (Table 8). Unfortunately, the baselines are not evaluated with all the metrics and datasets in their respective publications. Applying FFNN, OPHELIA outperforms baselines on the BBC dataset, and stays competitive on the AG News dataset, and on the Reuters dataset. We believe this behavior happens because of the size and configuration of the datasets. In BBC dataset, the texts of each category are very distinct. Meanwhile, the texts in the AG news, mainly the ones from the business and science/technology categories, present a similar pattern and confuse our models.

FFNNs are known for their ability to extract relevant features from the input data effectively. In the context of text classification, we infer that the FFNN model successfully captured important patterns and representations in the text data, leading to better performance. Meanwhile, Capsule Networks, with their ability to model hierarchical relationships and capture spatial configurations, are generally suitable for more complex data such as images. However, in certain text classification tasks, the data may not exhibit the hierarchical structures that CapsNet is designed to handle. However, for better results, it is interesting to carry out experiments with a greater variation of the network parameters.

Therefore, most of our experimental results are competitive, despite using low-dimension embeddings and a straightforward neural network architecture like the FFNN. It suggests that

text classification using jointly trained word embeddings and knowledge embeddings is promising, though more experiments are still needed to evaluate our proposal with specific vocabulary and particular domains.

Finally, though our approach did not yield statistically significant gains on the results, in a traditional sense, it showed its ability to be efficiently trained within a simple notebook setup. This computational efficiency underscores the practicality and accessibility of our approach, making it a valuable asset for scenarios where resource constraints and ease of implementation are paramount. While statistical significance may be a key metric in many contexts, the computational relevance of our approach positions it as a valuable tool for real-world applications where quick prototyping and implementation are critical.

Accuracy F1@Micro F1@Macro	BBC	AG News	Reuters
	-	-	-
Lenc et al. 2017 (LENC; KRÁL, 2017)	-	-	-
	-	-	**0.875**
	-	0.928	-
Qiao et al. 2018 (QIAO et al., 2018)	-	-	-
	-	-	-
	-	-	
Sinoara et al. 2019 (SINOARA et al., 2019)	0.979	-	-
	0.980	-	-
	-	-	**0.902**
Liu et al. 2019 (LIU et al., 2019)	-	-	0.903
	-	-	-
	-	0.864	-
Meng et al. 2020 (MENG et al., 2020)	-	-	-
	-	-	-
	-	0.918	-
Lee et al. 2021 (LEE; LEE; YU, 2021)	-	-	-
	-	-	-
	-	-	-
Liu et al. 2021 (LIU et al., 2021)	-	-	**0.908**
	-	-	0.675
	0.976	-	0.749
Pittaras et al. 2021 (PITTARAS et al., 2021)	-	-	-
	0.976	-	0.378
	-	**0.947**	-
Zhang et al. 2021 (ZHANG; YAMANA, 2021)	-	-	-
	-	-	-
	0.914	-	-
Shah et al. 2023 (SHAH et al., 2023)	-	-	-
	-	-	-
	0.985	0.919	0.869
OPHELIA FFNN	**0.984**	**0.918**	0.868
	0.982	**0.918**	0.718
	0.978	0.910	0.778
OPHELIA CapsNet	0.977	0.910	0.777
	0.976	0.910	0.544

Table 8 – Accuracy, Micro F1, and Macro F1 on the tested datasets.

6 CONCLUSION

The existing approaches for text classification have not fully exploited all the benefits of semantic resources like KGs and embeddings. An approach that combines word and knowledge embeddings has the potential to yield better results. This thesis proposes a text classification approach that employs joint embeddings of knowledge and words, taking advantage of what each one offers, with the goal of generating better results than the existing approaches.

We propose and develop OPHELIA, a neural network text classification approach based on joint embeddings of words and knowledge. OPHELIA can outperform and stays competitive with state-of-the-art approaches from the literature in news datasets with open domains. The neural network architecture of OPHELIA is relatively simple if compared with other architectures. Moreover, the best results were obtained with embeddings of only 50 dimensions, meaning that we can train our model in a single GPU or in hardware with relatively low RAM memory. Thus, OPHELIA has the potential to produce better results with a more sophisticated DNN architecture and a more significant training set.

This thesis contributes to the state-of-art by providing: (i) a comprehensive survey of text classification approaches that use embedding as a feature; (ii) a reference approach for text classification using text and KG embedding; (iii) a text classification system called OPHELIA based on the proposal; and (iv) publications related to this thesis. Besides comparing several text classification approaches that use embedding as a feature, the survey presents a decision tree that helps future readers decide which one is more suitable for their needs. Moreover, it presents the links to available source code repositories of open-source approaches. The general process and the reference approach are high-level and can be adapted to different needs. OPHELIA is a text classification system for news articles that different applications can use.

Future work for improving OPHELIA includes: (i) applying better preprocessing methods to distinct corpora and KGs to jointly train word and knowledge embeddings that may improve classification performance; (ii) using large language models such as BERT and GPT to generate contextualized embeddings and compare the results with those obtained by using fastText; (iii) making OPHELIA's NN-model interpretable by using current algorithms for interpreting black-box models and understanding how the model handles incorrect cases.

We also expect future works to analyze the multilingual aspect of feature representations and the impact of word sense and knowledge embeddings in text classification. Then, certain NLP tasks, such as entity recognition and linking (extraction and disambiguation of mentions to people, companies, locations, events, etc.), could be used for linking textual mentions to specific KG entities, relations and/or their embeddings.

In addition, successful approaches to classify texts in domains explored with more frequency, like news, could be evaluated and adapted to more challenging domains. New approaches could tackle challenges such as multiple terms used for referring to an entity; noisy text (i.e., with typos, grammatical errors, slang); and lack of contextual information of texts like those of short documents (e.g., social media posts). Lastly, joint classification of multi-modal

information, such as text or speech accompanied by images, is a challenge with many potential applications (e.g., detecting diverse categories of hate speech) due to the common use of hypermedia in mass communication nowadays. This challenge may be addressed by exploiting recent or promised breakthroughs in technologies like unified models for multi-modal information and knowledge.

BIBLIOGRAPHY

AGGARWAL, A.; SINGH, J.; GUPTA, K. A review of different text categorization techniques. **International Journal of Engineering and Technology(UAE)**, v. 7, p. 11–15, 07 2018.

AL-ANZI, F. S.; ABUZEINA, D. A micro-word based approach for arabic sentiment analysis. In: IEEE. **2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA)**. [S.l.], 2017. p. 910–914.

ALMEIDA, F.; XEXÉO, G. Word embeddings: A survey. **arXiv preprint arXiv:1901.09069**, 2019.

ALTINEL, B.; GANIZ, M. C. Semantic text classification: A survey of past and recent advances. **Information Processing & Management**, Elsevier, v. 54, n. 6, p. 1129–1153, 2018.

ALY, R.; REMUS, S.; BIEMANN, C. Hierarchical multi-label classification of text with capsule networks. In: **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop**. [S.l.: s.n.], 2019. p. 323–330.

AUBAID, A. M.; MISHRA, A. A rule-based approach to embedding techniques for text document classification. **Applied Sciences**, Multidisciplinary Digital Publishing Institute, v. 10, n. 11, p. 4009, 2020.

AUER, S. et al. Dbpedia: A nucleus for a web of open data. In: **The semantic web**. Berlin, Heidelberg: Springer, 2007. p. 722–735.

BAHDANAU, D.; CHO, K.; BENGIO, Y. Neural machine translation by jointly learning to align and translate. **CoRR**, abs/1409.0473, 2015.

BAKAROV, A. A survey of word embeddings evaluation methods. **arXiv preprint arXiv:1801.09536**, 2018.

BEBIS, G.; GEORGIOPOULOS, M. Feed-forward neural networks. **IEEE Potentials**, IEEE, v. 13, n. 4, p. 27–31, 1994.

BENGIO, Y. et al. A neural probabilistic language model. **Journal of machine learning research**, v. 3, n. Feb, p. 1137–1155, 2003.

BHATIA, K. et al. Sparse local embeddings for extreme multi-label classification. In: **Advances in Neural Information Processing Systems**. s.l.: Curran Associates, Inc., 2015. v. 29, p. 730–738.

BOJANOWSKI, P. et al. Enriching word vectors with subword information. **Trans. of the Association for Computational Linguistics**, v. 5, p. 135–146, 2017.

BORDES, A. et al. Translating embeddings for modeling multi-relational data. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2013. p. 2787–2795.

BOUNABI, M.; MOUTAOUAKIL, K. E.; SATORI, K. Neural embedding & hybrid ml models for text classification. In: IEEE. **2020 1st Intl. Conf. on Innovative Research in Applied Science, Engineering and Technology (IRASET)**. [S.l.], 2020. p. 1–6.

CAI, L. et al. A hybrid bert model that incorporates label semantics via adjustive attention for multi-label text classification. **Ieee Access**, IEEE, v. 8, p. 152183–152192, 2020.

CHALKIDIS, I. et al. Large-scale multi-label text classification on EU legislation. In: **57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers**. s.l.: Association for Computational Linguistics, 2019. p. 6314–6322.

CHANG, W.-C. et al. Taming pretrained transformers for extreme multi-label text classification. In: **26th ACM SIGKDD Intl. Conf. on Knowledge Discovery & Data Mining**. [S.l.: s.n.], 2020. p. 3163–3171.

COSTA, L. S. da; OLIVEIRA, I. L.; FILETO, R. Text classification using embeddings: a survey. **Knowledge and Information Systems**, Springer, p. 1–43, 2023.

CUI, P. et al. A survey on network embedding. **IEEE Trans. on Knowledge & Data Engineering**, IEEE Computer Society, PP, n. 01, p. 1–1, 2018.

CUNNINGHAM, P.; DELANY, S. J. k-nearest neighbour classifiers. **Multiple Classifier Systems**, Springer-Verlag, v. 34, n. 8, p. 1–17, 2007.

DENG, X. et al. Feature selection for text classification: A review. **Multimedia Tools and Applications**, Springer, v. 78, n. 3, p. 3797–3816, 2019.

DEVLIN, J. et al. BERT: Pre-training of deep bidirectional transformers for language understanding. In: **Conf. of the North American Chapter of the ACL**. s.l.: Association for Computational Linguistics (ACL), 2019. p. 4171–4186.

DIAB, M. S.; HAMAYDEH, M. N. K. Optimizing support vector machine classification based on semantic-text knowledge enrichment. **Palestinian Journal of Technology and Applied Sciences (PJTAS)**, v. 2, 2019.

DOMINGOS, P.; PAZZANI, M. On the optimality of the simple bayesian classifier under zero-one loss. **Machine learning**, Springer, v. 29, n. 2-3, p. 103–130, 1997.

DREYFUS, G. **Neural networks: methodology and applications**. [S.l.]: Springer Science & Business Media, 2005.

FABIAN, M.; GJERGJI, K.; GERHARD, W. Yago: A core of semantic knowledge unifying wordnet and wikipedia. In: **16th Intl. World Wide Web Conf., WWW**. [S.l.: s.n.], 2007. p. 697–706.

FIGUEIREDO, F. et al. Word co-occurrence features for text classification. **Information Systems**, Elsevier, v. 36, n. 5, p. 843–858, 2011.

GALLO, I. et al. Visual word embedding for text classification. In: SPRINGER. **Pattern Recognition. ICPR International Workshops and Challenges: Virtual Event, January 10–15, 2021, Proceedings, Part VI**. [S.l.], 2021. p. 339–352.

GAO, B. et al. Hierarchical taxonomy preparation for text categorization using consistent bipartite spectral graph copartitioning. **IEEE Transactions on Knowledge and Data Engineering**, IEEE, v. 17, n. 9, p. 1263–1273, 2005.

GARGIULO, F. et al. Deep neural network for hierarchical extreme multi-label text classification. **Applied Soft Computing**, Elsevier, v. 79, p. 125–138, 2019.

GOYAL, P.; FERRARA, E. Graph embedding techniques, applications, and performance: A survey. **Knowledge-Based Systems**, Elsevier, v. 151, p. 78–94, 2018.

GROSMAN, J. S. et al. Eras: Improving the quality control in the annotation process for natural language processing tasks. **Information Systems**, Elsevier, v. 93, p. 101553, 2020.

GUO, B. et al. Improving text classification with weighted word embeddings via a multi-channel textcnn model. **Neurocomputing**, Elsevier, v. 363, p. 366–374, 2019.

GUO, N. et al. Multi-level topical text categorization with wikipedia. In: **Proceedings of the 9th International Conference on Utility and Cloud Computing**. [S.l.]: Association for Computing Machinery, 2016. p. 343–352.

GUPTA, V. et al. Improving document classification with multi-sense embeddings. In: IEEE. **24th European Conference on Artificial Intelligence - ECAI**. Santiago de Compostela, Spain, 2020. p. 1–8.

HAVRLANT, L.; KREINOVICH, V. A simple probabilistic explanation of term frequency-inverse document frequency (tf-idf) heuristic (and variations motivated by this explanation). **International Journal of General Systems**, Taylor & Francis, v. 46, n. 1, p. 27–36, 2017.

HINTON, G. E.; KRIZHEVSKY, A.; WANG, S. D. Transforming auto-encoders. In: SPRINGER. **International conference on artificial neural networks**. [S.l.], 2011. p. 44–51.

HINTON, G. E.; SABOUR, S.; FROSST, N. Matrix capsules with em routing. In: **International conference on learning representations**. [S.l.: s.n.], 2018. p. 1–15.

HOSSAIN, M. R.; HOQUE, M. M.; SARKER, I. H. Text classification using convolution neural networks with fasttext embedding. In: ABRAHAM, A. et al. (Ed.). **Hybrid Intelligent Systems**. Cham: Springer International Publishing, 2021. p. 103–113.

HU, S. et al. Enhanced word embedding method in text classification. In: IEEE. **2020 6th Intl. Conf. on Big Data and Information Analytics (BigDIA)**. [S.l.], 2020. p. 18–22.

JIANG, T. et al. Lightxml: Transformer with dynamic negative sampling for high-performance extreme multi-label text classification. In: **The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)**. [S.l.: s.n.], 2021. p. 7987–7994.

JIN, P. et al. Bag-of-embeddings for text classification. In: **25th Intl. Joint Conf. on Artificial Intelligence**. s.l.: AAAI Press, 2016. (IJCAI'16, v. 16), p. 2824–2830.

JOHNSON, R.; ZHANG, T. Effective use of word order for text categorization with convolutional neural networks. **arXiv preprint arXiv:1412.1058**, 2014.

JOHNSON, R.; ZHANG, T. Semi-supervised convolutional neural networks for text categorization via region embedding. **Advances in neural information processing systems**, v. 28, 2015.

JONES, K. S. A statistical interpretation of term specificity and its application in retrieval. **Journal of documentation**, MCB UP Ltd, v. 28, n. 1, p. 11–21, 1972.

JOULIN, A. et al. Fasttext.zip: Compressing text classification models. **arXiv preprint arXiv:1612.03651**, 2016.

JOULIN, A. et al. Bag of tricks for efficient text classification. In: **Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers**. [S.l.]: Association for Computational Linguistics, 2017. p. 427–431.

JOULIN, A. et al. Fast linear model for knowledge graph embeddings. **arXiv preprint arXiv:1710.10881**, 2017.

KATARYA, R.; ARORA, Y. Study on text classification using capsule networks. In: IEEE. **2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)**. [S.l.], 2019. p. 501–505.

KIM, J. et al. Text classification using capsules. **Neurocomputing**, Elsevier, v. 376, p. 214–221, 2020.

KIM, Y. Convolutional neural networks for sentence classification. In: **2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)**. Doha, Qatar: Association for Computational Linguistics, 2014. p. 1746–1751.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014.

KIRYAKOV, A. et al. Semantic annotation, indexing, and retrieval. **Journal of Web Semantics**, Elsevier, v. 2, n. 1, p. 49–79, 2004.

KÖHN, A. What's in an embedding? analyzing word embeddings through multilingual evaluation. In: **2015 Conference on Empirical Methods in Natural Language Processing**. [S.l.: s.n.], 2015. p. 2067–2073.

KORDE, V.; MAHENDER, C. N. Text classification and classifiers: A survey. **International Journal of Artificial Intelligence & Applications**, Academy & Industry Research Collaboration Center (AIRCC), v. 3, n. 2, p. 85, 2012.

KOWSARI, K. et al. Hdltex: Hierarchical deep learning for text classification. In: IEEE. **2017 16th IEEE Intl. Conf. on machine learning and applications (ICMLA)**. [S.l.], 2017. p. 364–371.

KOWSARI, K. et al. Text classification algorithms: A survey. **Information**, Multidisciplinary Digital Publishing Institute, v. 10, n. 4, p. 150, 2019.

KUMAR, S.; KAR, A. K.; ILAVARASAN, P. V. Applications of text mining in services management: A systematic literature review. **International Journal of Information Management Data Insights**, Elsevier, v. 1, n. 1, p. 100008, 2021.

KUMAR, V. et al. Multi-label classification using hierarchical embedding. **Expert Systems with Applications**, Elsevier, v. 91, p. 263–269, 2018.

LAI, S. et al. How to generate a good word embedding. **IEEE Intelligent Systems**, IEEE, v. 31, n. 6, p. 5–14, 2016.

LAI, S. et al. Recurrent convolutional neural networks for text classification. In: **Twenty-ninth AAAI conference on artificial intelligence**. [S.l.: s.n.], 2015. p. 2267–2273.

LE, Q.; MIKOLOV, T. Distributed representations of sentences and documents. **31st Intl. Conf. on Machine Learning (ICML)**, v. 4, 05 2014.

LEE, S.; LEE, D.; YU, H. Oommix:out-of-manifold regularization in contextual embedding space for text classification. In: **59th Annual Meeting of the ACL and the 11th Intl. Joint Conf. on Natural Language Processing**. s.l.: Association for Computational Linguistics (ACL), 2021. p. 590–599.

LEHMANN, J. et al. DBpedia - a crystallization point for the web of data. **Journal of Web Semantics**, Elsevier Science Publishers B. V., v. 7, n. 3, p. 154–165, 2009.

LENC, L.; KRÁL, P. Word embeddings for multi-label document classification. In: **Intl. Conf. Recent Advances in Natural Language Processing, RANLP 2017**. Varna, Bulgaria: INCOMA Ltd., 2017. p. 431–437.

LEWIS, D. et al. Reuters-21578. **Test Collections**, v. 1, p. 19, 1987.

LI, J.; REN, F. Creating a chinese emotion lexicon based on corpus ren-cecps. In: IEEE. **2011 IEEE Intl. Conf. on Cloud Computing and Intelligence Systems**. [S.l.], 2011. p. 80–84.

LI, Q. et al. A survey on text classification: From traditional to deep learning. **ACM Transactions on Intelligent Systems and Technology (TIST)**, ACM New York, NY, v. 13, n. 2, p. 1–41, 2022.

LI, Y.; YANG, T. Word embedding for understanding natural language: a survey. In: **Guide to Big Data Applications**. [S.l.]: Springer, 2018. p. 83–104.

LI, Y.; YE, M. A text classification model base on region embedding and lstm. In: **2020 6th Intl. Conf. on Computing and Artificial Intelligence**. [S.l.: s.n.], 2020. p. 152–157.

LIN, Y. et al. Learning entity and relation embeddings for knowledge graph completion. In: **AAAI**. [S.l.: s.n.], 2015. v. 15, p. 2181–2187.

LIU, H. et al. Multi-label text classification via joint learning from label embedding and label correlation. **Neurocomputing**, Elsevier, 2021.

LIU, N.; WANG, Q.; REN, J. Label-embedding bi-directional attentive model for multi-label text classification. **Neural Processing Letters**, Springer, v. 53, n. 1, p. 375–389, 2021.

LIU, Q. et al. Task-oriented word embedding for text classification. In: **27th Intl. Conf. on computational linguistics**. [S.l.: s.n.], 2018. p. 2023–2032.

LIU, W. et al. A< word, part of speech> embedding model for text classification. **Expert Systems**, Wiley Online Library, v. 36, n. 6, p. e12460, 2019.

LIU, W. et al. The emerging trends of multi-label learning. **IEEE Trans. on pattern analysis and machine intelligence**, IEEE, 2021.

MENG, Y. et al. Text classification using label names only: A language model self-training approach. In: **EMNLP**. s.l.: Association for Computational Linguistics, 2020. p. 9006–9017.

MIKOLOV, T. et al. Efficient estimation of word representations in vector space. In: **1st Intl. Conf. on Learning Representations (ICLR) Workshop Track**. [S.l.: s.n.], 2013.

MIKOLOV, T. et al. Distributed representations of words and phrases and their compositionality. In: BURGES, C. J. C. et al. (Ed.). **Advances in Neural Information Processing Systems**. s.l.: Curran Associates, Inc., 2013. v. 26.

MONEDERO, J. S. et al. Challenges in ordinal classification: artificial neural networks and projection-based methods. Universidad de Granada, 2014.

MOREO, A.; ESULI, A.; SEBASTIANI, F. Word-class embeddings for multiclass text classification. **Data Mining and Knowledge Discovery**, Springer, v. 35, n. 3, p. 911–963, 2021.

MORO, A.; RAGANATO, A.; NAVIGLI, R. Entity Linking meets Word Sense Disambiguation: a Unified Approach. **Transactions of the Association for Computational Linguistics (TACL)**, v. 2, p. 231–244, 2014.

NADEAU, D.; SEKINE, S. A survey of named entity recognition and classification. **Lingvisticae Investigationes**, John Benjamins publishing company, v. 30, n. 1, p. 3–26, 2007.

NAM, J.; MENCÍA, E. L.; FüRNKRANZ, J. All-in text: Learning document, label, and word representations jointly. In: **Thirtieth AAAI Conference on Artificial Intelligence**. Phoenix, Arizona: AAAI Press, 2016. (AAAI'16), p. 1948–1954.

NAVIGLI, R. Word sense disambiguation: A survey. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 41, n. 2, 2009.

NAVIGLI, R. Word sense disambiguation: A survey. **ACM computing surveys (CSUR)**, ACM New York, NY, USA, v. 41, n. 2, p. 1–69, 2009.

NAVLANI, A. Text analytics for beginners using nltk. **Recovered from https://www. datacamp. com/community/tutorials/text-analytics-beginners-nltk**, 2018.

NICKEL, M. et al. A review of relational machine learning for knowledge graphs. **IEEE**, v. 104, n. 1, p. 11–33, 2016.

NICKEL, M. et al. Holographic embeddings of knowledge graphs. In: **AAAI**. [S.l.: s.n.], 2016. p. 1955–1961.

NICKEL, M.; TRESP, V.; KRIEGEL, H.-P. A three-way model for collective learning on multi-relational data. In: **Proceedings of the 28th International Conference on International Conference on Machine Learning**. USA: Omnipress, 2011. p. 809–816.

OLIVEIRA, I. L. et al. Towards holistic entity linking: Survey and directions. **Inf. Syst.**, v. 95, p. 101624, 2021.

PAN, C. et al. Few-shot transfer learning for text classification with lightweight word embedding based models. **IEEE Access**, IEEE, v. 7, p. 53296–53304, 2019.

PAPPAS, N.; HENDERSON, J. Gile: A generalized input-label embedding for text classification. **Trans. of the Association for Computational Linguistics**, v. 7, p. 139–155, 2019.

PATRICK, M. K. et al. Capsule networks–a survey. **Journal of King Saud University-Computer and Information Sciences**, Elsevier, 2019.

PENNINGTON, J.; SOCHER, R.; MANNING, C. Glove: Global vectors for word representation. In: **2014 conference on empirical methods in natural language processing (EMNLP)**. [S.l.: s.n.], 2014. p. 1532–1543.

PIETRAMALA, A. et al. A genetic algorithm for text classification rule induction. In: SPRINGER. **Joint European Conference on Machine Learning and Knowledge Discovery in Databases**. [S.l.], 2008. p. 188–203.

PITTARAS, N. et al. Text classification with semantically enriched word embeddings. **Natural Language Engineering**, Cambridge University Press, v. 27, n. 4, p. 391–425, 2021.

PRABHU, Y.; VARMA, M. Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning. In: **20th ACM SIGKDD Intl. Conf. on Knowledge discovery and data mining**. [S.l.: s.n.], 2014. p. 263–272.

QIAO, C. et al. A new method of region embedding for text classification. In: **Intl. Conf. on Learning Representations (Poster)**. [S.l.: s.n.], 2018. p. 1–12.

QUINLAN, J. R. Induction of decision trees. **Machine learning**, Springer, v. 1, n. 1, p. 81–106, 1986.

RYDNING, D. R.-J. G.-J.; REINSEL, J.; GANTZ, J. The digitization of the world from edge to core. **Framingham: International Data Corporation**, v. 16, 2018.

SABOUR, S.; FROSST, N.; HINTON, G. E. Dynamic routing between capsules. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2017. p. 3856–3866.

SARASWAT, A.; ABHISHEK, K.; KUMAR, S. Text classification using multilingual sentence embeddings. In: **Evolution in Computational Intelligence**. s.l.: Springer, 2021. p. 527–536.

SCHUTZE, H.; MANNING, C. D.; RAGHAVAN, P. **Introduction to information retrieval**. [S.l.]: Cambridge University Press, 2008.

SEBASTIANI, F. Machine learning in automated text categorization. **ACM computing surveys (CSUR)**, ACM, v. 34, n. 1, p. 1–47, 2002.

SHAH, M. A. et al. An automated text document classification framework using bert. **International Journal of Advanced Computer Science and Applications**, Science and Information (SAI) Organization Limited, v. 14, n. 3, 2023.

SHEN, W.; WANG, J.; HAN, J. Entity linking with a knowledge base: Issues, techniques, and solutions. **IEEE Trans. on Knowledge and Data Engineering**, v. 27, n. 2, p. 443–460, 2015.

SHEN, W.; WANG, J.; HAN, J. Entity linking with a knowledge base: Issues, techniques, and solutions. **IEEE Transactions on Knowledge and Data Engineering**, IEEE, v. 27, n. 2, p. 443–460, 2015.

SHI, M.; WANG, K.; LI, C. A c-lstm with word embedding model for news text classification. In: IEEE. **2019 IEEE/ACIS 18th Intl. Conf. on Computer and Information Science (ICIS)**. [S.l.], 2019. p. 253–257.

SINOARA, R. A. et al. Knowledge-enhanced document embeddings for text classification. **Knowledge-Based Systems**, Elsevier, v. 163, p. 955–971, 2019.

STEIN, R. A.; JAQUES, P. A.; VALIATI, J. F. An analysis of hierarchical text classification using word embeddings. **Information Sciences**, Elsevier, v. 471, p. 216–232, 2019.

TAILLÉ, B. Contextualization and generalization in entity and relation extraction. **arXiv preprint arXiv:2206.07558**, 2022.

TRIPATHI, N.; OAKES, M.; WERMTER, S. A scalable meta-classifier combining search and classification techniques for multi-level text categorization. **International Journal of Computational Intelligence and Applications**, World Scientific, v. 14, n. 04, p. 1550020, 2015.

VAPNIK, V. N. An overview of statistical learning theory. **IEEE transactions on neural networks**, Citeseer, v. 10, n. 5, p. 988–999, 1999.

VASWANI, A. et al. Attention is all you need. **Advances in neural information processing systems**, v. 30, 2017.

WALLACH, H. M. Topic modeling: beyond bag-of-words. In: **Proceedings of the 23rd international conference on Machine learning**. [S.l.: s.n.], 2006. p. 977–984.

WANG, C. et al. Text classification with heterogeneous information network kernels. In: **Thirtieth AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2016.

WANG, G. et al. Joint embedding of words and labels for text classification. In: **56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**. Melbourne, Australia: Association for Computational Linguistics, 2018. p. 2321–2331.

WANG, Q. et al. Knowledge graph embedding: A survey of approaches and applications. **IEEE Trans. on Knowledge and Data Engineering**, IEEE, v. 29, n. 12, p. 2724–2743, 2017.

WANG, W. et al. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. **Advances in Neural Information Processing Systems**, v. 33, p. 5776–5788, 2020.

WANG, Y.; CUI, L.; ZHANG, Y. Using dynamic embeddings to improve static embeddings. **CoRR**, abs/1911.02929, 2019.

WANG, Z. et al. Knowledge graph and text jointly embedding. In: **Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)**. [S.l.: s.n.], 2014. p. 1591–1601.

WETZKER, R.; ZIMMERMANN, C.; BAUCKHAGE, C. Analyzing social bookmarking systems: A del. icio. us cookbook. In: **ECAI Mining Social Data Workshop**. [S.l.: s.n.], 2008. p. 26–30.

WU, L. et al. Word mover's embedding: From word2vec to document embedding. **Conference on Empirical Methods in Natural Language Processing**, Association for Computational Linguistics, Brussels, Belgium, p. 4524–4534, 2018.

XU, H. et al. Text classification with topic-based word embedding and convolutional neural networks. In: **7th ACM Intl. Conf. on Bioinformatics, Computational Biology, and Health Informatics**. [S.l.: s.n.], 2016. p. 88–97.

YANG, P. et al. SGM: sequence generation model for multi-label classification. In: **27th Intl. Conf. on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018**. [S.l.: s.n.], 2018. p. 3915–3926.

YANG, Y.; CHUTE, C. G. An example-based mapping method for text categorization and retrieval. **ACM Transactions on Information Systems (TOIS)**, ACM, v. 12, n. 3, p. 252–277, 1994.

YANG, Z. et al. Xlnet: Generalized autoregressive pretraining for language understanding. **Advances in neural information processing systems**, Curran Associates Inc., v. 32, 2019.

ZHANG, C.; YAMANA, H. Improving text classification using knowledge in labels. In: **2021 IEEE 6th Intl. Conf. on Big Data Analytics (ICBDA)**. [S.l.: s.n.], 2021. p. 193–197.

ZHANG, J.; LERTVITTAYAKUMJORN, P.; GUO, Y. Integrating semantic knowledge to tackle zero-shot text classification. In: **2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)**. Minneapolis, Minnesota: Association for Computational Linguistics, 2019. p. 1031–1040.

ZHANG, X.; ZHAO, J.; LECUN, Y. Character-level convolutional networks for text classification. **Advances in neural information processing systems**, v. 28, 2015.

ZHANG, Y.; JIN, R.; ZHOU, Z.-H. Understanding bag-of-words model: a statistical framework. **International Journal of Machine Learning and Cybernetics**, Springer, v. 1, n. 1, p. 43–52, 2010.

ZHAO, W. et al. Towards scalable and reliable capsule networks for challenging nlp applications. **arXiv preprint arXiv:1906.02829**, 2019.

ZHAO, W. et al. Investigating capsule networks with dynamic routing for text classification. In: **2018 conference on empirical methods in natural language processing**. [S.l.: s.n.], 2018. p. 3110–3119.

ZHONG, H. et al. Aligning knowledge and text embeddings by entity descriptions. In: **Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing**. [S.l.: s.n.], 2015. p. 267–272.

ZHU, G.; IGLESIAS, C. A. Exploiting semantic similarity for named entity disambiguation in knowledge graphs. **Expert Systems with Applications**, Elsevier, v. 101, p. 8–24, 2018.