Journal of Information Technology and Computing
https://jitc.sabapub.com
ISSN: 2709-5916

2023 Volume 4, Issue 1 : 1 – 14
DOI : 10.48185/jitc.v4i1.612

Ontology based Feature Selection and Weighting for Text classification using Machine Learning

Djelloul BOUCHIHA[1,*], Abdelghani BOUZIANE [1], Noureddine DOUMI [2]

[1]Dept. Mathematics and Computer Science, Ctr Univ Naama, EEDIS Lab., UDL-SBA, Algeria
[2]Department of Computer Science, Faculty of Technologies, University of Saida, Algeria

Received: 28.02.2023 • Accepted: 05.03.2023 • Published: 27.06.2023 • Final Version: 27.06.2023

Abstract: Text classification consists in attributing text (document) to its corresponding class (category). It can be performed using an artificial intelligence technique called machine learning. However, before training the machine learning model that classifies texts, three main steps are also mandatory: (1) Preprocessing, which cleans the text; (2) Feature selection, which chooses the features that significantly represent the text; and (3) Feature weighting, which aims at numerically representing text through feature vector. In this paper, we propose two algorithms for feature selection and feature weighting. Unlike most existing works, our algorithms are sense-based since they use ontology to represent not the syntax but the sense of a text as a feature vector. Experiments show that our approach gives encouraging results compared to existing works. However, some additional suggested improvements can make these results more impressiveText classification consists in attributing text (document) to its corresponding class (category). It can be performed using an artificial intelligence technique called machine learning. However, before training the machine learning model that classifies texts, three main steps are also mandatory: (1) Preprocessing, which cleans the text; (2) Feature selection, which chooses the features that significantly represent the text; and (3) Feature weighting, which aims at numerically representing text through feature vector. In this paper, we propose two algorithms for feature selection and feature weighting. Unlike most existing works, our algorithms are sense-based since they use ontology to represent not the syntax but the sense of a text as a feature vector. Experiments show that our approach gives encouraging results compared to existing works. However, some additional suggested improvements can make these results more impressive.

Keywords: Text Classification, Feature selection, Feature weighting, Machine Learning (ML), Ontology, WordNet

1. Introduction

Text classification, also called text categorization, aims to classify texts (documents) into specific classes (categories) [1]. Thus, text classification allows attributing a text to one predefined class. It finds applications in several fields, notably information retrieval (IR) [2], information filtering (IF) [3], Web filtering [4], email or spam filtering [5], news filtering [6], sentiment analysis [7], knowledge management (KM) [8], text summarization [9], etc.

Text classification issue can be addressed using an artificial intelligence technique called machine learning. Arthur Samuel defines Machine Learning (ML) as the research field that gives machines the capability to learn without explicit programming [10]. Before launching the ML algorithm, the

* Corresponding Author: djelloul.bouchiha@univ-sba.dz

classification process needs: (1) Preprocessing, which cleans the text; (2) Feature selection, which chooses the features (generally words) that significantly represent the text; (3) Feature weighting, which aims at numerically representing text through feature vector.

Generally, preprocessing is a common step for all the classifiers. However, several feature selection and weighting methods, and various ML algorithms can constitute a classifier. In this paper, we opted for a standard preprocessing step, and we used a classical ML algorithm, namely SVM. However, we propose new algorithms for feature selection and feature weighting. Both of them are ontology-based (sense-based). The first one takes all concepts and relations of all the domain ontologies that correspond to the text categories, and considers their features. The second algorithm builds the feature vector of a text by computing the number of terms (words) which are semantically close to each feature according to a similarity measure. The lack of domain ontologies has obstructed our solution. So, domain ontologies have been replaced by WordNet, which is a large lexical database that provides senses of English words [11]. Nouns, verbs, adverbs and adjectives are grouped into cognitive synonyms called synsets; each describes a distinct concept. Synsets are connected through lexical and conceptual-semantic relations. Nouns and verbs are organized into is-a or hypernym hierarchies.

The remainder of the paper is organized as follows: Section 2 reviews some feature selection and weighting methods; Section 3 describes the text classification process; Section 4 presents our proposed text classification system, notably feature selection and weighting algorithms; Section 5 is devoted to experiments, where classification tool, evaluation and comparative study are discussed; and finally, Section 6 gives some conclusions and perspectives.

2. Related work

A feature in a text can be a simple term (word), complex linguistic structure (e.g. part of speech (POS)), supported information (e.g. word's first position), statistical structure (e.g. n-gram), Named Entity (e.g., person's name), etc. [12]. Feature selection consists in selecting a subset of the features that describe the texts. In the literature, feature selection is also referred to as Dimensionality Reduction, because it aims to reduce the feature matrix's dimensions that will be defined later. Feature selection should increase the classification accuracy and decrease the computational complexity (time and space) by deleting noise features. Thus, the feature selection step is important to improve the text classifier's accuracy, efficiency and scalability [13].

Next is a non-exhaustive list of feature selection techniques:

- **Information gain (IG)** has the same statistical meaning as the Kullback–Leibler divergence [14]. In text classification, IG is frequently used as a goodness criterion of a term (word). It looks for the term in a document to compute the number of bits of information, then predicts the class (category) of this document [15].
- **Chi-square (also called Chi-squared test or $\chi 2$ test):** originally, Pearson published a paper on Chi-square where he investigated a test of goodness of fit [16]. To classify texts, Chi-square is employed to measure the relevance between t (term) and C (class) [17]. Galavotti et al. proposed a simplified variant of CHI-square called GSS Coefficient (GSS) [18]. The authors in [19] proposed another variant of Chi-Square, called Correlation coefficient (CC) or NGL.
- **Mutual information (MI)** was defined and analyzed for the first time by Claude Shannon [20]. However, he did not call it Mutual Information. This term appears later in [21]. So, MI measures the mutual dependence between two random variables in probability theory and information theory. To select features from a text, MI measures the variations in the distribution of terms, and attributes the much higher rank to the terms of a positive nature [22].

- **Term strength (TS)** was initially proposed and evaluated for vocabulary reduction in text retrieval [23]. Later, TS was used in text classification [24, 25]. In this context, TS estimates the importance of a term based on how many times this term usually appears in closely-related texts. A "training set" is used to get pairs whose cosine similarity is greater than a threshold. Then, TS is calculated by considering the estimated conditional probability that a term appears in the 2nd half of a pair of related texts, given that it appears in the 1st half.
- **Odds ratios (OR)** is a statistical measure of the association between exposure and outcome [26]. In text classification, OR was proposed to select terms with relevant feedback [27]. OR starts from the fact that the distribution of features on the relevant documents is different from the distribution of features on the non-relevant documents. Mladenić, in [27], defines three measures inspired by the original OR formula: FreqOddsRatio, FreqLogP and ExpP.
- **Gini Index (GI)** measures the purity of the features concerning the class [28]. In text classification, purity is the discrimination level of a term to distinguish between possible classes.
- **Term Frequency (TF) and Document Frequency (DF):** TF is defined as the number of times a term occurs in a text. TF can be used to select features from a text. For example, we keep only features (terms) where TF exceeds a threshold. DF is the number of documents (texts) in which a term occurs. DF can also be used to select features from a text. For example, we maintain only features (terms) for which DF exceeds a threshold.

Most feature selection techniques cited above are word-based (term-based). The advantage of word-based techniques is that a large text is reduced to a set of simple independent terms, making the classification efficient. However, relationships between terms are lost [29]. Besides this, Semantic ambiguity (polysemy and synonymy issues) occurs when using terms as features [12]. To overcome these problems, an ontology-based (sense-based) feature selection method should be used.

Gruber defines ontology as an explicit specification of a conceptualization [30]. In computer science, ontology is the working model of entities and interactions [31]. The ontology consists mainly of concepts, relations, instances and axioms. The concepts correspond to a set of entities or things within a domain. The relations describe the interactions between the concepts. The instances are the things represented by a concept. The axioms are used to constrain values for concepts or instances. To the best of our knowledge, no work actually uses the domain ontology for feature selection and weighting in the text classification process. However, few attempts use WordNet in this context.

In [32], the authors used a dataset (Brown Corpus semantic concordance) annotated with WordNet to compare word-based and sense-based features in the text classification process. With a small training set (182 texts), they didn't significantly improve the classification effectiveness with sense-based features. In [33], the author proposed sequence kernels for words and POS Tags, which detect basic syntactic information and basic lexical semantics. Moschitti concluded that his kernels are more effective and efficient than previous models. Peng & Choi in [34] proposed text classification based on the words' senses and the relationships between the senses. Their experiment showed that using WordNet semantic hierarchy to have a sense-based document representation increases classification accuracy.

After the feature selection step, a text will be represented as a feature vector that consists of weights of features in the considered text. The feature weight indicates the degree of importance of the feature in the text; it can be represented, for example, by the feature occurrences in the text. Feature weighting of a set of texts generates feature vectors that constitute the so-called feature matrix. This matrix is of dimensions m*n, with m the number of texts, and n the number of features. In the literature, Feature weighting is also referred to as Feature extraction, Indexing or Document representation.

Several feature weighting techniques can be found in the literature:

- **Bag-of-Words (BoW)** appeared earlier in a linguistic context [35]. Lately, it has been widely applied in text classification [36], where each word's frequency (occurrence) is used as a feature value for training a classifier. So, in this method, a text is represented as a bag (a set) of words, disregarding the grammar and order of the words, but keeping only their multiplicities.
- **N-gram** is a set of n words appearing in a document in that order [37]. N-gram was introduced in a mathematical theory of communication [20]. It was later used in Natural Language Processing (NLP), where N-gram is usually defined as a sequence of N words [38].
- **Term Frequency - Inverse Document Frequency (TFIDF)** computes the importance of a word within a text (document) [39]. TFIDF is the product of Term Frequency (TF) and Inverse Document Frequency (IDF). TF was introduced as the first form of term weighting, the weight of a term that occurs in a text [40]. IDF quantified the specificity of a term by the inverse function of the number of texts in which the term appears [41].
- **Word2vec** is an NLP technique that uses neural networks to learn relationships between words from a huge dataset [42, 43]. As a result, each word is represented by a list of numbers called a vector such that the semantic similarity between two words corresponds to the similarity between their vectors. An extension of wor2vec, called doc2vec, aims to represent a word as vector, and the entire document as a vector [44].
- **HashingVectorizer** converts a collection of texts into a matrix of token occurrences [45]. The text vectorizer implementation uses the hashing trick [46] to convert a token (string) into a feature (integer).

3. Text classification Process

Our text classification process starts with a dataset consisting of a set of English texts that belong to several categories. Since we opted, in this paper, for supervised learning, we used a labeled dataset, i.e. each text must be annotated with its corresponding category. The dataset will be split into two parts: a training set (70% of the dataset) and a test set (30% of the dataset). As shown in Figure 1, the text classification process consists of three main phases: Training, Test and Prediction.

The **training phase** receives as input Training set that undergoes four main stages:

1. **Preprocessing** receives as input English texts, cleans these texts, and generates tokenized cleaned texts [47].
2. **Feature selection** selects a subset of the features available for describing the texts to reduce the dimensions of the so called feature matrix; lines of the feature matrix are feature vectors of texts. This step produces a set of selected features.
   Note that Feature selection was drawn in Figure 1 with dashed lines because it is sometimes not specified in the classification process. In this case, all the tokens (words) from the preprocessing step are considered features and will be weighted in the next step.
3. **Feature weighting** generates a numerical representation of cleaned texts. As a result, this step generates a feature matrix.
4. **Machine learning algorithm** builds a machine learning model that constitutes the kernel of a classifier. The model represents what was learned by the machine learning algorithm.

The **test phase** consists in running the classifiers generated from the previous phase on the test set and measuring each classifier's performance. As an output of this phase, the best classifier will be selected.

Finally, **the prediction phase** receives a new English text introduced by the user as input. Then, the best classifier determines the text category.

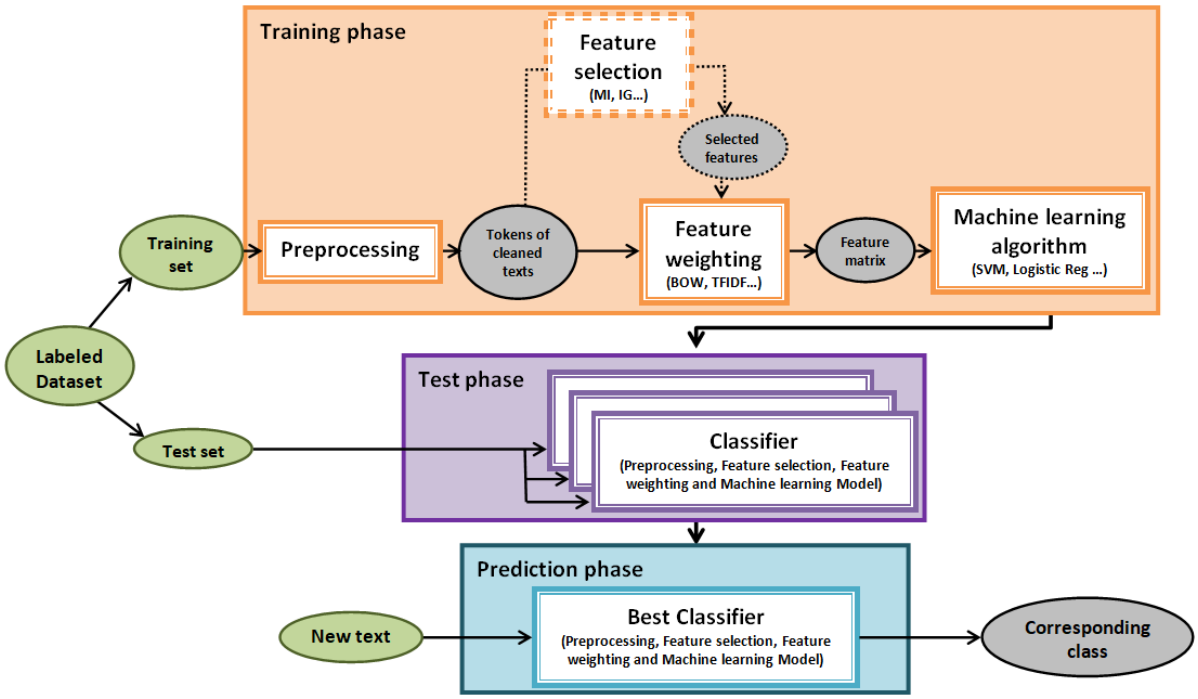


Figure 1. Text classification process

4. Proposed text classification system

In this paper, we have set up a system that covers the four steps of the classification process:

4.1. Preprocessing

Preprocessing includes tokenization and normalization. It also removes stop-words, numbers, punctuations, links, white-spaces and non-English words.

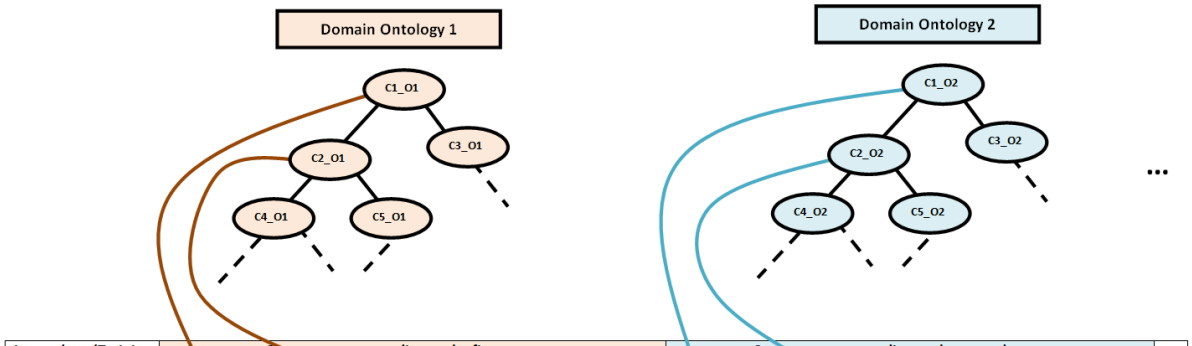
Tokenization is splitting a text into subunits called tokens [48]. Generally, a token is a word of the text.

Normalization is reducing a token to its base form. Two normalization techniques are usually used: stemming and lemmatization [49].

Stop-words are high-frequency words in a document, such as "the", "but" and "not" that are filtered out, because their presence in a text fails to distinguish it from the other texts [50]. Thus, stop-words do not contribute to the content of the text [36], and consequently, they are deleted from the text. In addition, numbers, punctuation, URI links, multiple white-spaces and non-English words are removed from the text because they have no impact on the classification process.

4.2. Ontology based feature selection approach

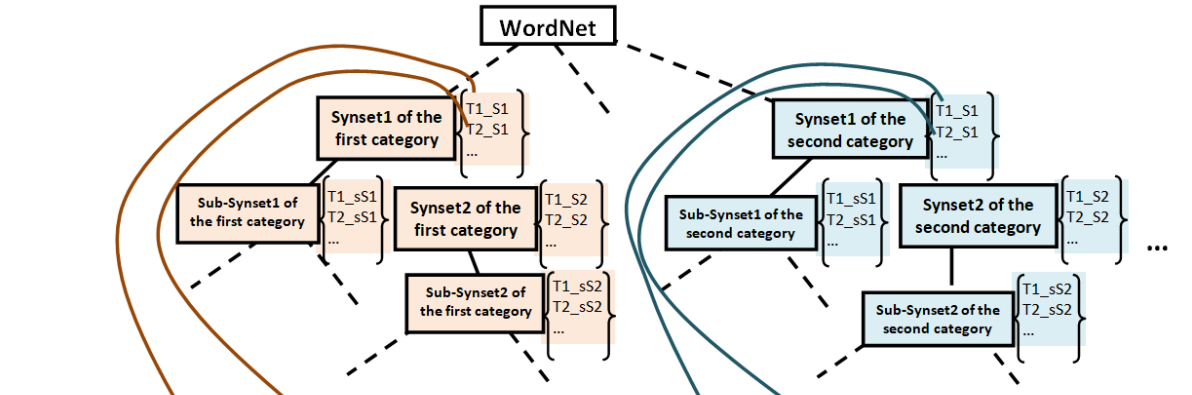
In this paper, we propose an ontology based feature selection algorithm. As shown in Figure 2, our first idea was to take a set of domain ontologies; each corresponds to a category of texts. Then, we consider all the concepts and relations of these ontologies as the selected features.



Input data (Training set, Test set or new input Text)	Concepts corresponding to the first category				Concepts corresponding to the second category				...
	Concept1_onto1	Concept2_Onto1	...	Conceptn_Onto1	Concept1_Onto2	Concept2_Onto2	...	Conceptm_Onto2	...
Text1	Number of times C1_O1 appears in Text1	Number of times C2_O1 appears in Text1	...	Number of times Cn_O1 appears in Text1	Number of times C1_O2 appears in Text1	Number of times C2_O2 appears in Text1	...	Number of times Cm_O2 appears in Text1	...
Text2	Number of times C1_O1 appears in Text2	Number of times C2_O1 appears in Text2	...	Number of times Cn_O1 appears in Text2	Number of times C1_O2 appears in Text2	Number of times C2_O2 appears in Text2	...	Number of times Cm_O2 appears in Text2	...
...	...	...	...	...	...	...	...	...	...

Figure 2. Ontology based feature selection and weighting

The lack of domain ontologies has obstructed this first idea. So, domain ontologies have been replaced by WordNet as illustrated in Figure 3. In this case, selected features are all the terms of all synsets (and their hyponyms) that correspond to the texts' categories.



Input data (Training set, Test set or new input Text)	Terms of synsets corresponding to first category				Terms of synsets corresponding to the second category				...
	Term1_Synset1	Term2_Synset1	...	Termn_Synsetx	Term1_Synset1	Term2_Synset1	...	Termm_Synsety	...
Text1	Number of times T1_S1 appears in Text1	Number of times T2_S1 appears in Text1	...	Number of times Tn_Sx appears in Text1	Number of times T1_S1 appears in Text1	Number of times T2_S1 appears in Text1	...	Number of times Tm_Sy appears in Text1	...
Text2	Number of times T1_S1 appears in Text2	Number of times T2_S1 appears in Text1	...	Number of times Tn_Sx appears in Text2	Number of times T1_S1 appears in Text2	Number of times T2_S1 appears in Text2	...	Number of times Tm_Sy appears in Text2	...
...	...	...	...	...	...	...	...	...	...

Figure 3. WordNet based feature selection and weighting

Next is the first proposed algorithm for feature selection:

Algorithm 1: Feature Selection
Inputs: Categories, WordNet
Outputs: Features_list
Begin
For each Category
From WordNet, extracting all synsets where the Category name occurs
For each extracted synset
Add all the terms of the synset to Features_list
From WordNet, extracting all hyponyms of the synset
For each hyponym
Add all the terms of the hyponym to Features_list

```
            End_for
        End_for
    End_for
End
```

The algorithm receives WordNet and a list of categories taken from the dataset as input. It returns a list of features that will be weighted in the next step.

4.3. Ontology based feature weighting approach

For the third step of the classification process, we proposed a second algorithm that builds the feature vector of a text by computing the number of terms (in the text) that are semantically close to each feature using similarity measures.

```
Algorithm 2: Feature weighting
    Inputs: Texts, Features_list, WordNet, Similarity, Threshold
    Output: Feature_Matrix
Begin
    For each Text
        For each Feature in Features_list
            For_each Term in Text
                If (Similarity(Term, Feature) >= Threshold):
                    Feature_Matrix[Text, Feature] = Feature_Matrix[Text, Feature] + 1
                End_if
            End_for
        End_for
    End_for
End
```

The algorithm receives as input: texts, features, WordNet, a similarity measure and a threshold, and returns a feature matrix.

A feature matrix is a set of feature vectors; each corresponding to a text. Similarity measure computes how much two elements are alike; it returns a value in [0..1]; the value 1 is given when the two elements are semantically equivalent. The threshold is the value that decides if a term and feature are semantically close so that they can be considered equivalent.

4.4. Machine learning algorithm

Several ML algorithms exist in the literature. In our classification system, we opted for Support Vector machines (SVM), a nonlinear generalization of the Generalized Portrait algorithm used initially for pattern recognition and computer learning [51, 52]. Afterward, SVM has been introduced explicitly as a new machine learning algorithm to resolve classification problems [53]. It maps the input vectors into high dimensional space and constructs separating hyper-plane(s).

5. Experiments

To perform our experiments, we first choose BBC dataset [54]. The BBC dataset consists of 2225 texts taken from the BBC news website distributed over 5 categories: politics, business, entertainment, sport and tech. It was randomly split into a training set (70%) and a test set (30%). Then, we opt for Python language to implement our classifier.

5.1. Implementation

Python is a powerful programming language [55]. It is also easy to learn. Python's library contains built-in modules that provide standardized solutions for many problems when implementing our classifier. A module is a file containing Python definitions and statements. A collection of modules is called a package. Python allows us to install other external packages, among which we cite:

- For the preprocessing step, the nltk [50], textblob [56] and tashaphyne [57] packages have to be installed.
- Feature selection and weighting need the installation of gensim [58]. The numpy [59] package is also necessary for these two stages.
- To implement ML algorithm, we used scikit-learn package [45, 60].

All these Python packages helped us implement our ontology based classifier that we made open access[1] for the benefit of the NLP and AI communities.

5.2. Evaluation

As mentioned above, we implemented our classification tool1 covering: preprocessing, ontology based feature selection and weighting, and SVM. Then, we checked our classification tool using the BBC dataset [54]. Our classifier dealt with 2225*226 feature matrix: 2225 is the number of documents in the BBC dataset, and 226 is the number of the terms of WordNet synsets that are semantically related to the five categories in the BBC dataset.

As an evaluation metric, we opted for Accuracy, which is the fraction of the study population that is decided correctly [61]. For a classification problem, Accuracy = (P + N)/T, where P (true positives) is the number of documents correctly classified, N (true negatives) is the number of documents correctly not classified, and T is the total number of documents.

We recall that our proposed weighting algorithm (Section 4.3) needs two important inputs: threshold and similarity measure. The question that strongly arises now is: which similarity measure gives the best classification results, and with which threshold?

To answer this question, we implemented six WordNet based similarity measures. For each one, we gave a series of threshold values.

As shown in Figure 4, the implemented similarity measures relying on WordNet are: *path* [62], which is computed by inversing the length of the shortest path between two synsets; *lch* [63], which scales the shortest path between the two synsets by the maximum path length in the is–a hierarchy in which the synsets appear; *wup* [64], which computes the path length from the LCS (Least Common Subsumer) of the two synsets to the root node, and scales this value by the sum of the path lengths from the root to the individual synsets; *res* [65], which returns a score based on the LCS information content and that of the two synsets; both, *lin* [66] and *jcn* [67], increase the LCS information content by the sum of the information content of the two synsets; while *lin* scales the LCS information content by the sum, *jcn* subtracts the LCS information content from the sum, and then converts the inverse from a distance to a similarity measure.

Different combinations (Similarity - Threshold) have been tested. Our ontology based classifier reaches its best result (Accuracy of 0.82) with *wup* similarity and a threshold of 0.8.

[1] https://github.com/khouloud-1/Ontology_Based_Classifier

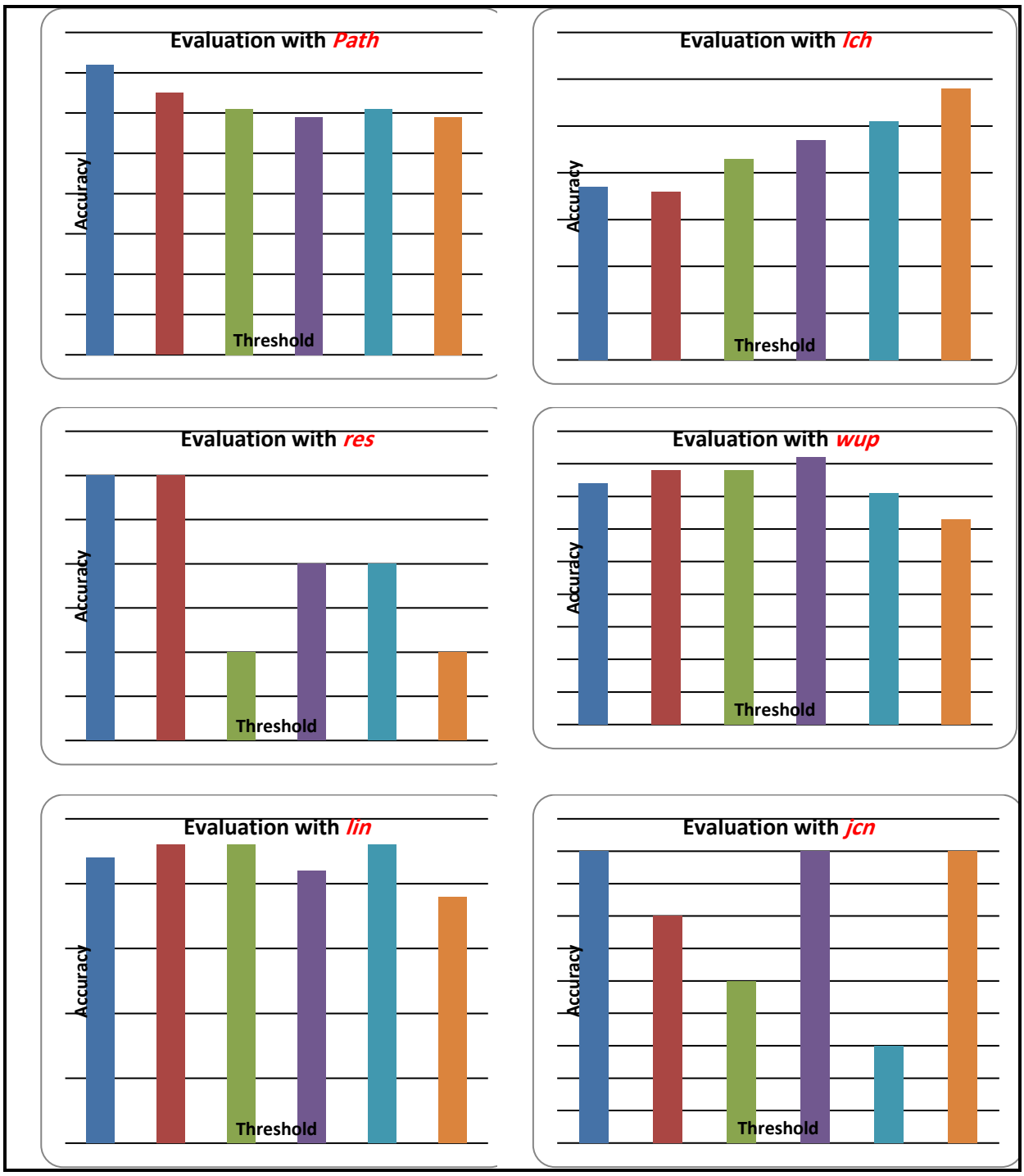


Figure 4. Evaluation results with different similarity measures

5.3. Comparative study

To show the efficiency of our classifier, we compare it to three other classifiers[2], all of them have the same preprocessing step, and the same ML algorithm, namely SVM. Also, the three classifiers have word-based feature selection step. However, the first classifier uses BoW for feature weighting, the second uses TFIDF, and the third uses Doc2Vec.

[2] https://github.com/khouloud-1/Ontology_Based_Classifier

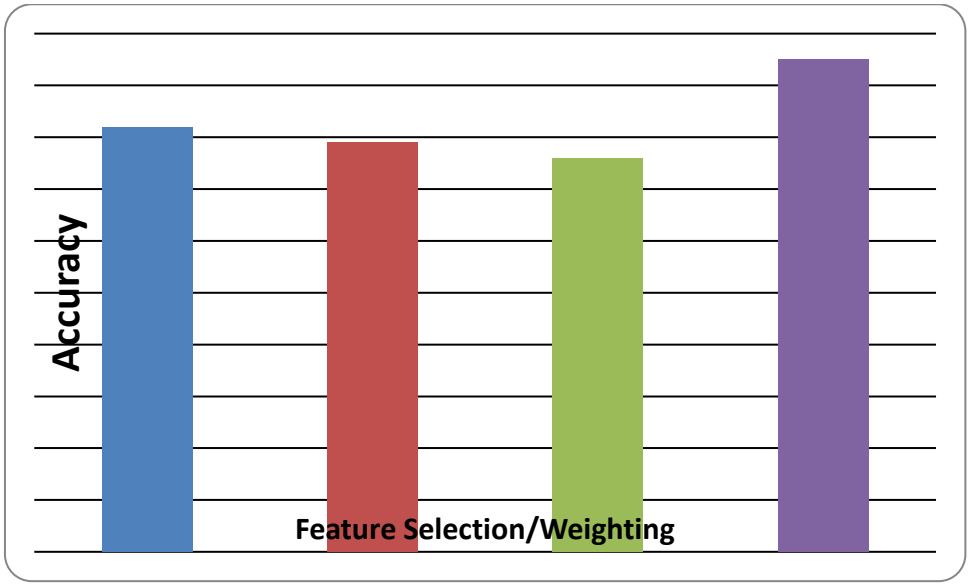


Figure 5. Classification Accuracy with different feature selection/weighting methods

Figure 5 shows that our approach outperforms two classifiers. However, it needs more improvement to be the best one. This can be justified by the fact that, in its current version, our WordNet based classifier uses only the Nouns hierarchy. Besides this, WordNet is a lexical ontology dedicated mainly to linguistic applications.

We think that using actual domain ontologies instead of lexical WordNet, will improve our classification system's result for many reasons:

- Concepts of the ontology are specific Nouns describing the category to which a text belongs.
- Relations between concepts will represent Verbs in a text.
- Ontology instances can be Named Entities in the text.
- Ontology axioms can infer hidden information that can contribute to the classification process.

6. Conclusion and perspectives

Feature selection and weighting are primordial steps in the text classification process aiming to attribute a text to its corresponding category. While feature selection extracts important features from a text, feature weighting represents the text through a feature vector that includes a set of values; each represents the weight (importance degree) of the feature in the text. To be accomplished, the text classification process also needs Preprocessing as the first step, and ML algorithm as the last one.

In this context, most existing classifiers use word-based feature selection and weighting techniques. In this paper, we propose two sense-based algorithms for selecting and weighting features. Both consider concepts of the domain ontologies as the features that can characterize a text to be classified. Our first intention was to use domain ontologies corresponding to the texts categories. However, we substitute them by WordNet due to the lack of such ontologies.

A classification tool has been implemented, and experiments have been conducted to show the efficiency of our approach compared to the existing works. Experiment results were encouraging; however, some additional suggested improvements can make these results more impressive.

As future work, we plan to use real domain ontologies, or RDF Linked Data, which is data interlinked with other data [68], widely available on the Web, like Dbpedia [69].

In its current version, our classifier uses only two kinds of WordNet similarity measures in the second algorithm: path-based (*wup*, *lch* and *path*) and content-based (*jcn*, *lin* and *res*). The next version should implement relation-based measures to improve the classification results. Relation-based measures are: *hso* [70], which computes the relatedness between two synsets by looking for a path between them that isn't too long and that doesn't change direction too often; *lesk* [71], which computes relatedness by scoring the overlaps between the synsets' glosses; and *vector* [72], which computes relatedness by finding the cosine between the gloss vectors of the two synsets.

References

[1] K. Nalini and L. J. Sheela, "Survey on text classification," *International Journal of Innovative Research in Advanced Engineering*, vol. 1, pp. 412-417, 2014.

[2] W. B. Croft, D. Metzler, and T. Strohman, *Search engines: Information retrieval in practice* vol. 520: Addison-Wesley Reading, 2010.

[3] C. Lanquillon, "Enhancing text classification to improve information filtering," Otto-von-Guericke-Universität Magdeburg, Universitätsbibliothek, 2001.

[4] R. Du, R. Safavi-Naini, and W. Susilo, "Web filtering using text classification," in *The 11th IEEE International Conference on Networks, 2003. ICON2003.*, Sydney, NSW, Australia, 2003, pp. 325-330.

[5] A. Bhowmick and S. M. Hazarika, "E-Mail Spam Filtering: A Review of Techniques and Trends," in *Advances in Electronics, Communication and Computing*, A. Kalam, S. Das, and K. Sharma, Eds., ed Singapore: Springer Singapore, 2018, pp. 583-590.

[6] K. Lang, "NewsWeeder: Learning to Filter Netnews," in *Machine Learning Proceedings 1995*, A. Prieditis and S. Russell, Eds., ed San Francisco (CA): Morgan Kaufmann, 1995, pp. 331-339.

[7] B. Liu and L. Zhang, "A Survey of Opinion Mining and Sentiment Analysis," in *Mining Text Data*, C. C. Aggarwal and C. Zhai, Eds., ed Boston, MA: Springer US, 2012, pp. 415-463.

[8] M. Heidarysafa, K. Kowsari, L. Barnes, and D. Brown, "Analysis of Railway Accidents' Narratives Using Deep Learning," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Orlando, Florida, USA, 2018, pp. 1446-1453.

[9] I. Mani and M. T. Maybury, *Advances in Automatic Text Summarization*, abridged, illustrated, reprint ed.: MIT Press, 1999.

[10] A. L. Samuel, "Some Studies in Machine Learning Using the Game of Checkers," *IBM Journal of Research and Development*, vol. 3, pp. 210-229, 1959.

[11] G. A. Miller, "WordNet: a lexical database for English," *Commun. ACM*, vol. 38, pp. 39–41, 1995.

[12] X. Zhou, R. Gururajan, Y. Li, R. Venkataraman, X. Tao, G. Bargshady, P. D. Barua, and S. Kondalsamy-Chennakesavan, "A survey on text classification and its applications," *Web Intelligence*, vol. 18, pp. 205-216, 2020.

[13] J. Chen, H. Huang, S. Tian, and Y. Qu, "Feature selection for text classification with Naïve Bayes," *Expert Systems with Applications*, vol. 36, pp. 5432-5435, 2009/04/01/ 2009.

[14] S. Kullback and R. A. Leibler, "On Information and Sufficiency," *The Annals of Mathematical Statistics*, vol. 22, pp. 79-86, 1951.

[15] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81-106, 1986/03/01 1986.

[16] K. Pearson, "X. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 50, pp. 157-175, 1900/07/01 1900.

[17] Y. Zhai, W. Song, X. Liu, L. Liu, and X. Zhao, "A Chi-Square Statistics Based Feature Selection Method in Text Classification," in *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, Beijing, China, 2018, pp. 160-163.

[18] L. Galavotti, F. Sebastiani, and M. Simi, "Experiments on the Use of Feature Selection and Negative Evidence in Automated Text Categorization," in *Proceedings of the 4th European Conference on Research and Advanced Technology for Digital Libraries*, Lisbon, Portugal, 2000, pp. 59-68.

[19] H. T. Ng, W. B. Goh, and K. L. Low, "Feature selection, perceptron learning, and a usability case study for text categorization," in *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, Philadelphia, Pennsylvania, USA, 1997, pp. 67–73.

[20] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal,* vol. 27, pp. 379-423, 1948.

[21] R. M. Fano, "Transmission of Information: A Statistical Theory of Communications," *American Journal of Physics,* vol. 29, pp. 793-794, 1961.

[22] D. Agnihotri, K. Verma, and P. Tripathi, "Mutual information using sample variance for text feature selection," in *Proceedings of the 3rd International Conference on Communication and Information Processing*, Tokyo, Japan, 2017, pp. 39–44.

[23] W. J. Wilbur and K. Sirotkin, "The automatic identification of stop words," *Journal of information science,* vol. 18, pp. 45-55, 1992.

[24] Y. Yang, "Noise reduction in a statistical approach to text categorization," in *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, Seattle, Washington, USA, 1995, pp. 256–263.

[25] Y. Yang and J. Wilbur, "Using corpus statistics to remove redundant words in text categorization," *Journal of the American Society for Information Science,* vol. 47, pp. 357-369, 1996.

[26] M. Szumilas, "Explaining odds ratios," *Journal of the Canadian academy of child and adolescent psychiatry,* vol. 19, pp. 227-229, 2010.

[27] D. Mladenić, "Feature subset selection in text-learning," in *Machine Learning: ECML-98*, Berlin, Heidelberg, 1998, pp. 95-100.

[28] W. Shang, H. Huang, H. Zhu, Y. Lin, Y. Qu, and Z. Wang, "A novel feature selection algorithm for text categorization," *Expert Systems with Applications,* vol. 33, pp. 1-5, 2007/07/01/ 2007.

[29] D. Shen, J.-T. Sun, Q. Yang, H. Zhao, and Z. Chen, "Text Classification Improved through Automatically Extracted Sequences," in *Proceedings of the 22nd International Conference on Data Engineering (ICDE'06)*, Atlanta, GA, USA, 2006, pp. 121-121.

[30] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge Acquisition,* vol. 5, pp. 199-220, 1993/06/01/ 1993.

[31] R. Stevens, C. A. Goble, and S. Bechhofer, "Ontology-based knowledge representation for bioinformatics," *Briefings in Bioinformatics,* vol. 1, pp. 398-414, 2000.

[32] A. Kehagias, V. Petridis, V. G. Kaburlasos, and P. Fragkou, "A Comparison of Word- and Sense-Based Text Categorization Using Several Classification Algorithms," *Journal of Intelligent Information Systems,* vol. 21, pp. 227-247, 2003/11/01 2003.

[33] A. Moschitti, "Syntactic and semantic kernels for short text pair categorization," in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, Athens, Greece, 2009, pp. 576–584.

[34] X. Peng and B. Choi, "Document Classifications based on Word Semantic Hierarchies," in *Proceedings of the International Conference on Artificial Intelligence and Applications (AIA'05)*, Innsbruck, Austria, 2005, pp. 362-367.

[35] Z. S. Harris, "Distributional Structure," *WORD,* vol. 10, pp. 146-162, 1954/08/01 1954.

[36] M. F. McTear, Z. Callejas, and D. Griol, *The conversational interface*, 1 ed. vol. 6: Springer Cham, 2016.

[37] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text Classification Algorithms: A Survey," *Information,* vol. 10, p. 150, 2019.

[38] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, Third Edition draft ed., 2021.

[39] A. Rajaraman and J. D. Ullman, *Mining of Massive Datasets*: Cambridge University Press, 2011.

[40] H. P. Luhn, "A Statistical Approach to Mechanized Encoding and Searching of Literary Information," *IBM Journal of Research and Development,* vol. 1, pp. 309-317, 1957.

[41] K. Sparck Jones, "A STATISTICAL INTERPRETATION OF TERM SPECIFICITY AND ITS APPLICATION IN RETRIEVAL," *Journal of Documentation,* vol. 28, pp. 11-21, 1972.

[42] T. Mikolov, K. Chen, G. Corrado, and J. Dean. (2013, October 2022). Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781. Available: https://ui.adsabs.harvard.edu/abs/2013arXiv1301.3781M

[43] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in neural information processing systems,* vol. 26, 2013.

[44] Q. Le and T. Mikolov, "Distributed Representations of Sentences and Documents," in *Proceedings of the 31st International Conference on Machine Learning*, Beijing China, 2014, pp. 1188--1196.

[45] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. Vanderplas, A. Joly, B. Holt, and G. Varoquaux. (2013, October 2022). API design for machine learning software: experiences from the scikit-learn project. arXiv:1309.0238. Available: https://ui.adsabs.harvard.edu/abs/2013arXiv1309.0238B

[46] K. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg, "Feature hashing for large scale multitask learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*, Montreal, Quebec, Canada, 2009, pp. 1113–1120.

[47] D. Bouchiha, A. Bouziane, and N. Doumi, "Machine Learning for Arabic Text Classification: A Comparative Study," *Malaysian Journal of Science and Advanced Technology*, vol. 2, pp. 163-173, 2022.

[48] G. Grefenstette, "Tokenization," in *Syntactic Wordclass Tagging*, H. van Halteren, Ed., ed Dordrecht: Springer Netherlands, 1999, pp. 117-133.

[49] M. Toman, R. Tesar, and K. Jezek, "Influence of word normalization on text classification," in *Proceeding of Multidisciplinary Approaches to Global Information Systems, InSciT 2006*, Merida, Spain, 2006, pp. 354-358.

[50] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*: O'Reilly Media, 2009.

[51] V. Vapnik and A. Chervonenkis, "A note on one class of perceptrons," *Automation and Remote Control,* vol. 25, pp. 821-837, 1964.

[52] V. Vapnik and A. Lerner, "Pattern recognition using generalized portrait method," *Automation and Remote Control,* vol. 24, pp. 774-780, 1963.

[53] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning,* vol. 20, pp. 273-297, 1995/09/01 1995.

[54] D. Greene and P. Cunningham, "Practical solutions to the problem of diagonal dominance in kernel document clustering," in *Proceedings of the 23rd international conference on Machine learning*, Pittsburgh, Pennsylvania, USA, 2006, pp. 377–384.

[55] P. S. Foundation. (2022, October 2022). *Python 3.10.7 documentation*. Available: https://docs.python.org/3/

[56] S. Loria. (2020, October 2022). textblob Documentation. *Release 0.16.0*. Available: https://buildmedia.readthedocs.org/media/pdf/textblob/latest/textblob.pdf

[57] T. Zerrouki. (2019, October 2022). *Tashaphyne, Arabic light stemmer*. Available: https://pypi.org/project/Tashaphyne/0.3.4.1/

[58] R. Rehurek and P. Sojka, "Software framework for topic modelling with large corpora," in *Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks*, Valletta, Malta, 2010, pp. 45-50.

[59] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, pp. 357-362, 2020/09/01 2020.

[60] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[61] C. E. Metz, "Basic principles of ROC analysis," *Seminars in Nuclear Medicine*, vol. 8, pp. 283-298, 1978/10/01/ 1978.

[62] T. Pedersen, S. Patwardhan, and J. Michelizzi, "WordNet:: Similarity-Measuring the Relatedness of Concepts," in *Proceedings of the Nineteenth National Conference on Artificial Intelligence (Sponsored by the AAAI)*, San Jose, California, USA, 2004, pp. 25-29.

[63] C. Leacock and M. Chodorow, "Combining local context and WordNet similarity for word sense identification," in *WordNet: An electronic lexical database*. vol. 49, ed, 1998, pp. 265-283.

[64] Z. Wu and M. Palmer, "Verbs semantics and lexical selection," in *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, Las Cruces, New Mexico, USA, 1994, pp. 133–138.

[65] P. Resnik, "Using information content to evaluate semantic similarity in a taxonomy," in *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, Montreal, Quebec, Canada, 1995, pp. 448-453.

[66] D. Lin, "An Information-Theoretic Definition of Similarity," in *Proceedings of the Fifteenth International Conference on Machine Learning*, 1998, pp. 296–304.

[67] J. J. Jiang and D. W. Conrath, "Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy," in *Proceedings of the 10th Research on Computational Linguistics International Conference*, Taipei, Taiwan, 1997, pp. 19-33.

[68] T. Berners-Lee. (2006, October 2022). *Linked data-design issues*. Available: https://www.w3.org/DesignIssues/LinkedData.html

[69] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. van Kleef, S. Auer, and C. Bizer, "DBpedia – A large-scale, multilingual knowledge base extracted from Wikipedia," *Semantic Web*, vol. 6, pp. 167-195, 2015.

[70] G. Hirst and D. St-Onge, "Lexical chains as representations of context for the detection and correction of malapropisms," in *WordNet: An electronic lexical database*. vol. 305, ed: MIT Press, 1998, pp. 305-332.

[71] S. Banerjee and T. Pedersen, "Extended gloss overlaps as a measure of semantic relatedness," in *Proceedings of the 18th international joint conference on Artificial intelligence*, Acapulco, Mexico, 2003, pp. 805–810.

[72] S. Patwardhan, "Incorporating dictionary and corpus information into a context vector measure of semantic relatedness (Doctoral dissertation, University of Minnesota, Duluth)," 2003.