Artificial Intelligence and Law (2019) 27:199–225
https://doi.org/10.1007/s10506-018-9236-y

Unsupervised and supervised text similarity systems for automated identification of national implementing measures of European directives

Rohan Nanda[1] · **Giovanni Siragusa**[1] · **Luigi Di Caro**[1] · **Guido Boella**[1] · **Lorenzo Grossio**[2] · **Marco Gerbaudo**[2] · **Francesco Costamagna**[2]

Published online: 26 October 2018

Abstract
The automated identification of national implementations (NIMs) of European directives by text similarity techniques has shown promising preliminary results. Previous works have proposed and utilized unsupervised lexical and semantic similarity techniques based on vector space models, latent semantic analysis and topic models. However, these techniques were evaluated on a small multilingual corpus of directives and NIMs. In this paper, we utilize word and paragraph embedding models learned by shallow neural networks from a multilingual legal corpus of European directives and national legislation (from Ireland, Luxembourg and Italy) to develop unsupervised semantic similarity systems to identify transpositions. We evaluate these models and compare their results with the previous unsupervised methods on a multilingual test corpus of 43 Directives and their corresponding NIMs. We also develop supervised machine learning models to identify transpositions and compare their performance with different feature sets.

Keywords Text similarity · Transposition · Machine learning

1 Introduction

The European Union (EU) Member States are responsible for the correct and timely implementation of the EU legislation into national law. European directives in particular have to be transposed by adopting national implementing measures (NIMs). However, there have been many shortcomings in the implementation of European law over the years (Eliantonio et al. 2013). The European Commission (EC) as the guardian of the Treaties is responsible to ensure that the national law is compliant with the EU directives. Therefore, the Commission plays a critical role in

✉ Rohan Nanda
nanda@di.unito.it

Extended author information available on the last page of the article

monitoring the implementation of national law to ensure effective transposition of directives. After Member States have adopted the NIMs, the Commission starts monitoring them to ensure the correct transposition of the directive. The monitoring steps include the preparation of Conformity check reports and Correlation tables.[1] Conformity check reports comprise legal analysis and concordance tables. They are prepared by legal consulting firms for NIMs of different Member States. The concordance tables identify the implementing NIM provisions for each article of the directive in a tabular format. Correlation tables are quite similar to concordance tables but they are prepared by the Member States and sent confidentially to the Commission.

The manual monitoring steps taken by the Commission are quite laborious and expensive as they require thorough legal analysis (Ciavarini Azzi 2000). Further, it becomes even more difficult to monitor national implementations for cross-border legal research in different Member States. The EUR-Lex portal provides information about the NIMs for a particular directive at the level of legal acts. It does not identify the specific transposing provisions for a particular article of the directive. Legal experts involved in monitoring of EU directives need to identify the transposed provisions within the NIMs to correctly evaluate the transposition of the directive. In this paper, we develop and evaluate both unsupervised and supervised text similarity techniques to identify transpositions. We developed unsupervised semantic similarity models by learning word and paragraph vector representations from a combined corpus of European directives and national legislation (from Ireland, Luxembourg and Italy). These models were used to identify transpositions on a multilingual corpus of 43 directives and their corresponding NIMs from Ireland, Luxembourg and Italy. Our experiments show that paragraph vector model outperformed other word embedding-based models, such as word2vec and fastText. We also evaluated the performance of previous text similarity techniques used to identify transpositions such as latent semantic analysis, TF–IDF cosine, latent dirichlet allocation and unifying similarity measure (USM) on this corpus (Nanda et al. 2017a, 2016). The results show that the TF–IDF cosine based on the vector space model has a better performance than other unsupervised text similarity models. We compare the results from different unsupervised text similarity models and present their advantages and drawbacks to identify transpositions. We also implemented different supervised machine learning classifiers using the gold standard labelled data to identify transpositions. Our results indicate that support vector machine (SVM) classifier with TF–IDF features had the best performance to identify transpositions among the supervised methods.

The rest of the paper is organized as follows. In the next section, we present the related work. Section 3 describes the corpus and pre-processing pipeline. Section 4 presents the unsupervised text similarity models. The supervised text similarity models are discussed in Sect. 5. The paper concludes in Sect. 6.

[1] http://www.europarl.europa.eu/sides/getAllAnswers.do?reference=E-2010-9931&language=SL.

2 Related work

In this section, we discuss state-of-the-art methods for text similarity on legal texts. The first work in automated identification of national implementing measures (NIMs) utilized text similarity techniques based on vector space model, latent semantic analysis (LSA) and EuroVoc thesaurus[2] (Nanda et al. 2016). The text similarity methods were evaluated on a corpus of five directives and their corresponding NIMs for the English legislation (from Ireland and the United Kingdom). The results indicate that cosine similarity based on term frequency–inverse document frequency (TF–IDF) weighting scheme achieved the best F-score. The application of dimensionality reduction models such as LSA resulted in the loss of some essential features (in short texts) needed to capture semantic similarity. The authors also concluded that the addition of semantic knowledge from EuroVoc did not improve the performance of LSA and cosine similarity. In a recent work, Nanda et al. (2017a) proposed a unifying similarity measure (USM) for automated identification of NIMs. The model utilized features such as common words, common sequences of words and partial string matches. The system was evaluated on a small multilingual corpus to identify transpositions in English, French and Italian legislation. USM achieved a good performance across all three legislation and outperformed state-of-the-art methods for text similarity, such as LSA and latent dirichlet allocation (LDA). The French legislation (from Luxembourg) achieved the best F-score as compared to the English and Italian legislation. Humphreys et al. (2015) developed a system to map recitals to legal provisions in the European legislation. A gold standard mapping was developed to link the recitals in the preamble with the articles in the normative provisions. However, the authors did not include the mappings from recitals to sub-provisions. A cosine similarity score was computed between the TF–IDF recitals and provisions vectors. The results indicate that the system achieved a high accuracy due to the presence of a large number of true negatives (unbalanced dataset). The system achieved a high recall but with low precision. The system could be used to automatically identify all possible correspondences between recitals and provisions but they would need to be checked by a legal knowledge engineer. Boella et al. (2012) integrated a cosine similarity based measure into a legal knowledge management system (Boella et al. 2016) for identifying relevant legislative documents for a particular legislation. The system uses the class labels of legislative articles, related to the lightweight ontology used in the system, described in Ajani et al. (2017), along with the cosine similarity score to identify the most relevant legislative texts for a give legislation. Magerman et al. (2010) investigated the application of text similarity techniques based on vector space models and latent semantic analysis (LSA) to map patents and scientific publications. The system was evaluated on a corpus of 467 documents (30 patents and 437 publications). The pre-processing pipeline comprised stop-words removal, stemming, term reduction and weighting. The results indicate that the TF–IDF weighting scheme

[2] http://eurovoc.europa.eu.

using vector space models achieved the best performance. The authors investigated the application of LSA with different singular value decomposition (SVD) ranks for approximation. They inferred that for their small dataset higher values of SVD ranks perform better than low rank values. They also noticed the application of LSA transform over TF–IDF weights degrades the performance of the TF–IDF model. Mandal et al. (2017) utilized different similarity measures to identify similar court cases from the Indian Supreme Court. The legal case documents were utilized for text similarity by selecting four different representations: whole document, document summary, paragraphs and reason for citation (the text surrounding the citations to other cases). They implemented four models of document similarity: TF–IDF, word2vec, latent dirichlet allocation (LDA) and doc2vec (also known as paragraph vectors). The results demonstrate that doc2vec outperforms other models in case of whole document. This is because doc2vec is the only model in their implementation which captures the word order to some extent (Le and Mikolov 2014). In case of paragraphs, both word2vec and doc2vec have similar performance and outperform other methods. Overall their results indicate that the doc2vec similarity over the entire document has the highest semantic correlation with legal expert opinion. This was demonstrated by a higher pearson correlation coefficient of 0.69 in case of whole documents as compared to a 0.59 correlation coefficient in case of paragraphs. Aletras et al. (2016) developed a machine learning system to predict the judicial decisions of the European Court of Human Rights (ECHR). The system utilized textual features from different subsections of the case such as "relevant applicable law", "facts", "circumstances", "Law" and "full case" to predict whether there has been a violation of an article of the convention of human rights. A dataset of 584 cases was compiled from articles 3, 6 and 8 of the Convention. The authors utilized N-grams and topics as features for the binary classifier. The top-2000 most frequent N-grams (for $N \epsilon \{1, 2, 3, 4\}$) were utilized from the dataset. Topics were created for each article in the dataset by clustering semantically similar N-grams together. A support vector machine (SVM) classifier was trained using the textual features to predict if there is a violation or non-violation for a particular case (with respect to the Article of the Convention). The results indicate that N-gram features from the "circumstances" subsection achieve a better performance as compared to other subsections. The topics features developed by clustering similar N-grams achieve the highest accuracy from all the feature set. Topics capture the overall gist from the N-grams of different subsections and thus are able to be a good predictor. They also infer that the information contained in the "circumstance" subsection is a key predictor in determining if the case is a violation or not.

3 Corpus preparation and pre-processing

3.1 Corpus preparation

We prepared a multilingual parallel corpus of 43 directives and their corresponding NIMs for Ireland, Luxembourg and Italian legislation. Table 1 presents the CELEX numbers of the directives and NIMs as per EUR-Lex. Each legislative document was

Table 1 The CELEX numbers of directives and NIMs in the multilingual corpus

Sno	Directives	NIMs (Ireland)	NIMs (Luxembourg)	NIMs (Italy)
1	32010L0024	72010L0024IRL_188115	72010L0024LUX_194845	72010L0024ITA_195371
2	32009L0128	72009L0128IRL_190844	72009L0128LUX_222878	72009L0128ITA_195369
			72009L0128LUX_222460	
3	31994L0011	71994L0011IRL_97765	71994L0011LUX_97767	71994L0011ITA_97762
4	31996L0040	71996L0040IRL_103146	71996L0040LUX_103149	71996L0040ITA_103143
5	31996L0093	71996L0093IRL_104711	71996L0093LUX_104713	71996L0093ITA_104707
6	31997L0043	71997L0043IRL_106134	71997L0043LUX_106145	71997L0043ITA_106123
7	31998L0058	71998L0058IRL_107788	71998L0058LUX_107790	71998L0058ITA_107789
8	31998L0084	71998L0084IRL_108777	71998L0084LUX_108779	71998L0084ITA_108778
9	31999L0105	71999L0105IRL_111554	71999L0105LUX_126553	71999L0105ITA_111555
			71999L0105LUX_126554	
10	32009L0021	72009L0021IRL_184902	72009L0021LUX_189874	72009L0021ITA_186849
11	31999L0002	71999L0002IRL_109429	71999L0002LUX_109431	71999L0002ITA_109430
12	32009L0020	72009L0020IRL_188439	72009L0020LUX_189875	72009L0020ITA_194551
13	31999L0095	71999L0095IRL_111630	71999L0095LUX_111632	71999L0095ITA_125921
14	32009L0033	72009L0033IRL_183965	72009L0033LUX_183231	72009L0033ITA_179616
15	32000L0036	72000L0036IRL_112636	72000L0036LUX_112638	72000L0036ITA_112637
16	32000L0055	72000L0055IRL_113427	72000L0055LUX_113429	72000L0055ITA_113428
17	32001L0110	72001L0110IRL_116005	72001L0110LUX_116006	72001L0110ITA_30057
18	32008L0090	72008L0090IRL_168455	72008L0090LUX_168629	72008L0090ITA_170924
19	32001L0112	72001L0112IRL_116042	72001L0112LUX_116043	72001L0112ITA_29334
20	32001L0113	72001L0113IRL_116060	72001L0113LUX_116062	72001L0113ITA_116061
21	32007L0002	72007L0002IRL_170884	72007L0002LUX_170775	72007L0002ITA_167690
22	32007L0043	72007L0043IRL_170239	72007L0043LUX_170162	72007L0043ITA_173275
23	32007L0033	72007L0033IRL_170294	72007L0033LUX_170795	72007L0033ITA_173410
24	32001L0111	72001L0111IRL_116024	72001L0111LUX_116026	72001L0111ITA_116025
25	32005L0094	72005L0094IRL_142403	72005L0094LUX_131762	72005L0094ITA_167074
26	32001L0081	72001L0081IRL_115688	72001L0081LUX_115689	72001L0081ITA_29985
		72001L0081IRL_194972		
27	32001L0095	72001L0095IRL_28698	72001L0095LUX_135144	72001L0095ITA_29986
				72001L0095ITA_135265
28	32004L0023	72004L0023IRL_131105	72004L0023LUX_147977	72004L0023ITA_150656
				72004L0023ITA_150706
29	32001L0096	72001L0096IRL_115977	72001L0096LUX_115979	72001L0096ITA_35623
		72001L0096IRL_115978		
30	32002L0092	72002L0092IRL_34868	72002L0092LUX_126481	72002L0092ITA_125142
			72002L0092LUX_123898	
31	32003L0094	72003L0094IRL_33063	72003L0094LUX_33944	72003L0094ITA_132883
32	32014L0028	72014L0028IRL_239853	72014L0028LUX_243958	72014L0028ITA_237982
33	32015L0413	72015L0413IRL_250326	72015L0413LUX_234950	72015L0413ITA_214698
34	32013L0053	72013L0053IRL_245865	72013L0053LUX_243962	72013L0053ITA_233695
			72013L0053LUX_243961	72013L0053ITA_233693

Table 1 (continued)

Sno	Directives	NIMs (Ireland)	NIMs (Luxembourg)	NIMs (Italy)
35	32006L0088	72006L0088IRL_157218	72006L0088LUX_153017	72006L0088ITA_158323
36	32008L0057	72008L0057IRL_185250	72008L0057LUX_169960	72008L0057ITA_173702
37	32008L0096	72008L0096IRL_186546	72008L0096LUX_190526	72008L0096ITA_180588
				72008L0096ITA_180158
38	32008L0043	72008L0043IRL_161791	72008L0043LUX_161581	72008L0043ITA_166919
			72008L0043LUX_161580	
39	32005L0062	72005L0062IRL_137665	72005L0062LUX_129420	72005L0062ITA_150819
				72005L0062ITA_150669
				72005L0062ITA_150695
40	31999L0092	71999L0092IRL_111679	71999L0092LUX_120249	71999L0092ITA_111680
41	32001L0024	72001L0024IRL_180124	72001L0024LUX_114418	72001L0024ITA_30729
		72001L0024IRL_28393		
42	32002L0044	72002L0044IRL_133618	72002L0044LUX_142436	72002L0044ITA_124474
43	32003L0010	72003L0010IRL_133619	72003L0010LUX_142437	72003L0010ITA_132468

stored in a proprietary XML format with each XML element representing a legal provision (directive article or NIM provision). A gold standard mapping between directive articles and NIM provisions was prepared by two legal researchers with expertise in European law. An inter-annotator agreement was computed for each language corpus (of 43 directives and their corresponding NIMs) using Cohen's Kappa (McHugh 2012). The mean Kappa scores for English (from Ireland), French (from Luxembourg) and Italian (from Italy) corpus are 0.4812, 0.79 and 0.6065 respectively. This indicates that the agreement was highest in the Luxembourg Directive-NIM corpus and the lowest in Ireland Directive-NIM corpus. Due to the highly time-consuming and expensive process of preparing the gold standard mapping we did not include directives with a large number of NIMs. Further, we were also restricted in our choice of directives due to the fact that many directives did not have NIMs from all three Member states (Luxembourg, Ireland and Italy).

3.2 Pre-processing and vectorization

A multilingual NLP pipeline was developed for processing the corpus. The directive and NIM documents in XML format are processed to extract the legal provisions. Each legal provision is linked to a unique label (article or provision number). The next step involves pre-processing the text. Pre-processing helps in removing noise and generating a high quality representation of text for semantic similarity. First of all sentence tokenization is carried out to segment provisions into sentences. Then word tokenizers are used to extract words from sentences. The obtained tokens are converted into lowercase. We utilized spaCy's[3] list of stopwords for French and Italian to filter out

[3] https://spacy.io/.

common words in directives and NIMs. For English, we used NLTK's stopwords list (Bird and Loper 2004). The punctuation was also removed. The remaining tokens were tagged with part-of-speech (POS) tags (POS tag of a token is taken as an input by the lemmatizer to correctly lemmatize it). For English, we utilized NLTK's WordNet lemmatizer. For French and Italian we used spaCy's default lemmatizer. Our experiments in feature selection indicate that keeping only specific POS tags like nouns, verbs and adjectives lead to loss of essential features which are necessary for short text similarity. Other POS tags also contain important semantic information which must be preserved. Therefore, we do not filter out tokens for any particular POS tag. Each provision in the corpus is thus represented in a bag-of-words format. It is a list of each token and its count in a particular provision.

4 Unsupervised text similarity models to identify transpositions

In this section, we discuss the unsupervised text similarity models and their results on the multilingual corpus.

4.1 Lexical and semantic unsupervised text similarity models

In this section, we present the lexical and semantic unsupervised text similarity models. We also compare and analyze their results on the multilingual Directive-NIM corpus.

4.1.1 TF–IDF cosine

The output from Sect. 3.2 is a bag-of-words representation. A provision-term matrix is then constructed with a collection of all provision vectors in the corpus. The rows of the matrix consist of the terms and the columns correspond to the provisions. This representation of documents or provisions as vectors in a common vector space is called as vector space model (VSM). We applied Term Frequency–Inverse Document Frequency (TF–IDF) weighting method to the provision-term matrix (Sparck Jones 1972). The TF–IDF measure evaluates the importance of each term, by offsetting its frequency in the provision with its frequency in the corpus. The TF–IDF weight of term t in provision p is given as follows:

$$tf{-}idf_{t,p} = (tf_{t,p}) \cdot \log \frac{N}{\mathrm{pf}_t} \tag{1}$$

where $tf_{t,p}$ is the term frequency of term t in provision P, N is the number of provisions in the corpus and pf_t is the provision frequency of term t in the corpus. The cosine similarity measure between article vector A and provision vector P is computed as follows:

$$CS(A, P) = \frac{A \cdot P}{|A||P|} \tag{2}$$

The dot product of the article and provision vector is divided by the product of their lengths (lengths computed by Euclidean distance) to compute the cosine similarity.

4.1.2 Latent semantic analysis (LSA)

One of the major drawbacks of utilizing the vector space model (VSM) is its inability to deal with polysemy and synonymy. Latent Semantic Analysis (LSA) is a popular indexing method in information retrieval which is used to produce a low-rank approximation matrix for the document-term matrix (provision-term matrix in our case) by using word co-occurrence (Deerwester et al. 1990). The derived features of LSA have been shown to capture polysemy and synonymy to some extent (Deerwester et al. 1990). LSA uses singular value decomposition (SVD) to project the provision vectors into a reduced latent space (Golub and Reinsch 1970). SVD decomposes the provision-term matrix into separate matrices which capture the similarity between terms and provisions across different dimensions in space. The relationship between terms is represented in a subspace approximation of the original vector space to reduce noise and find latent relations between terms and documents. The original provision-term matrix X is reduced to a lower rank approximation matrix, X_k, where the rank k is much smaller than the original rank of matrix X. The approximation is represented as follows:

$$X_k = U \Sigma_k V^T \tag{3}$$

The $\sum$ matrix represents the singular values of X. U and V represent the left singular vector and right singular vector respectively. The truncated matrix $(V')^T$ represents the provisions in the reduced k-dimensional space. The query, A_i (directive article) is also transformed into the LSA space as follows:

$$A_{ik} = \Sigma_k^{-1} U_k^T A_i \tag{4}$$

The cosine similarity values are computed between the directive article and the corresponding NIM provisions to retrieve the most similar NIM provisions. We experimented with different number of latent dimensions on our dataset and the best performance was observed at 50 dimensions (chosen value for results).

4.1.3 Latent dirichlet allocation (LDA)

Latent Dirichlet Allocation (LDA) is a generative model which discovers a latent distribution of topics in a corpus of documents. It is based on the assumption that a document can be represented as a mixture of hidden topics (Blei et al. 2003). LDA is a probabilistic topic model characterized by a conditional word by document probability distribution, $p(w|d)$ (Bergamaschi and Po 2014). This distribution is a combination of topic by document distribution, p(z|d) and word by topic distribution, $p(w|z)$:

$$p(w|d) = \sum_z p(w|z)p(z|d) \tag{5}$$

Thus, each document d is represented as a multinomial distribution of latent topics z, and each topic z is represented as a multinomial distribution of words w. The LDA

transform is applied over the TF–IDF provision term matrix to obtain provision-topic matrix. Each provision vector is thus represented in a reduced dimension as a topic distribution. Our experiments with different number of topics suggested that LDA's performance improved with the increase in number of topics. We chose 500 topics for the LDA model.

4.1.4 Unifying similarity measure (USM)

A unifying similarity measure (USM) was proposed to identify the national implementations of EU directives (Nanda et al. 2017a). USM combines cosine similarity $CS(A, P)$, N-gram similarity $N(A,P)$ and approximate string matching $AS(A, P)$ methods using a weighted arithmetic mean as follows:

$$USM(A, P) = \frac{w_1 \times CS(A, P) + w_2 \times N(A, P) + w_3 \times AS(A, P)}{w_1 + w_2 + w_3} \tag{6}$$

where A is the directive article and P is the NIM provision. w_1, w_2 and w_3 are the weights assigned to cosine similarity, N-gram similarity and approximate similarity respectively. Inverse-variance weighting method was used to assign weights (Hartung et al. 2011). We implemented two variants of USM, USM_chars, with character N-grams and USM_tokens, with token N-grams. We utilized 4 g for both cases and N-gram similarity was computed as discussed in Nanda et al. (2017a).

4.1.5 Results of the lexical and semantic unsupervised text similarity models

In this section, we evaluate the models discussed in the above sections on the multilingual corpus of 43 directives and their corresponding NIMs. The system was evaluated by comparing the retrieved provisions with the gold standard mapping. The metrics precision, recall and F-score were computed for each directive by incrementing threshold values from 0 to 1 at intervals of 0.01. The threshold which provides the best F-score was chosen. We then computed the macro-average precision, recall and F-score metrics for each legislation corpus (Ireland, Luxembourg and Italy). The macro-average precision is computed by taking the average of the precision values for the 43 directives (for a particular legislation). The macro-average recall is computed by taking the average of the recall values for the 43 directives (for a particular legislation). The macro-average F-score is the harmonic mean of the macro-average precision and macro-average recall. Figure 1 presents the macro-average precision, recall and F-score of the lexical and semantic unsupervised text similarity models over the multilingual corpus. We observe that the Luxembourg Directive-NIM corpus achieves a higher precision, recall and F-score than the English and Italian corpus for each similarity measure. This is because of the presence of common words and phrases in European directives and the Luxembourg legislation. The Irish and Italian legislation had more linguistic variation with respect to the European directives. We consider article 4.2 of the directive (CELEX Number: 32013L0053) and its implementing NIM provisions for Ireland and Luxembourg legislation as per the gold standard

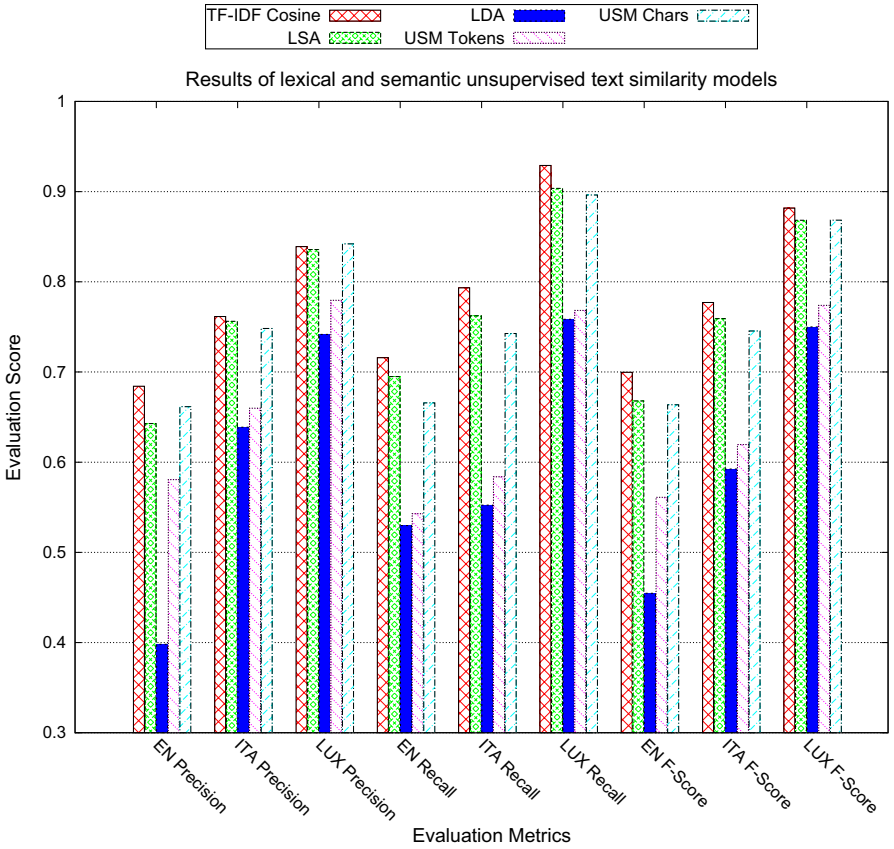

Fig. 1 Results of the lexical and semantic unsupervised text similarity models

Table 2 Article 4.2 of directive (CELEX Number: 32013L0053) and its implementing NIM provision 4.2 from Ireland legislation (CELEX Number: 72013L0053IRL_245865)

Article 4.2 of directive	Provision 4.2 of Ireland NIM
Member states shall ensure that the products referred to in article 2(1) are not made available on the market or put into service unless they comply with the requirements of paragraph 1	A person who makes available on the market a product to which these regulations apply in contravention of paragraph (1) shall be guilty of an offence

(Tables 2 and 3). In case of Ireland (Table 2), we notice that the article instructs the Member States to ensure that only the products that are compliant with the requirements of paragraph 1 should be made available on the market or put into service. The NIM provision on the other hand, explains the implications for a person who makes available on the market a product which violates the requirements

Table 3 Article 4.2 of directive (CELEX Number: 32013L0053) in French and its implementing NIM provision 4.2 from Luxembourg legislation (CELEX Number: 72013L0053LUX_243961)

Article 4.2 of directive	Provision 4.2 of Luxembourg NIM
Les États membres veillent à ce que les produits mentionnés à l'article 2, paragraphe 1, ne soient mis à disposition sur le marché ou mis en service que s'ils remplissent les critères du paragraphe 1	Le département de la surveillance du marché de l'ILNAS, désigné ci-après «le département de la surveillance du marché» veille à ce que les produits mentionnés à l'article 2, paragraphe 1er, ne soient mis à disposition sur le marché ou mis en service que s'ils remplissent les critères du paragraphe 1er

in paragraph 1. This illustrates that the NIM provision transposes the article by providing a specific legal implication (which was not mentioned in the directive article). We also observe that NIM provision does not mention the part about products being put into service. Therefore, these two provisions do not share a high magnitude of similarity. The lexical and semantic unsupervised text similarity techniques could not identify such cases of transposition. In case of Luxembourg (Table 3), the directive article has the same meaning as the English version. The NIM provision implements the article by explicitly specifying the authority name ("the Market Surveillance Department" in this case). However, the rest of the wordings are very similar to the directive article, which facilitates the identification of transposition by text similarity techniques. TF–IDF cosine similarity measure achieved the best F-score for all three corpora. The performance of LSA and USM_chars model was comparable and they were the second best methods after TF–IDF cosine in terms of F-score (Fig. 1). LSA has a slightly better performance (F-score) than USM_chars for English and Italian corpus. These results indicate that the application of dimensionality reduction techniques such as LSA and LDA do not improve the performance of the text similarity system. The idea behind such techniques is to reduce the variability in word usage and thus highlight the latent relations between words and documents which were obscured by noise (Cosma and Joy 2012). However, in case of short texts such as legal provisions, the reduction of dimensions results in loss of key features which maybe relevant for semantic similarity. This is also demonstrated in Italian and Luxembourg legislation corpus where LSA achieved a lower recall than TF–IDF cosine (Fig. 1). In terms of precision, the performance of LSA is almost equivalent to TF–IDF cosine (in Luxembourg and Italian legislation). The overall performance of LDA was poorer as compared to other methods. In case of short texts (such as tweets), they have been outperformed by TF–IDF based models (Hong and Davison 2010). USM_chars model had a decent performance over the multilingual corpus. There were some transpositions which were identified by USM_chars but missed by other methods. Table 4 presents one such example. It can be observed that the only similar part in directive article and NIM provision is about the road safety impact assessment being carried at the planning stage of the infrastructure project. The NIM provision then goes in further details which are not mentioned

Table 4 Article 3.2 of directive (CELEX Number: 32008L0096) and its implementing NIM provision 4.2 from Ireland legislation (CELEX Number: 72008L0096IRL_186546)

Article 3.2 of directive	Provision 4.2 of Ireland NIM
The road safety impact assessment shall be carried out at the initial planning stage before the infrastructure project is approved. In that connection, member states shall endeavour to meet the criteria set out in Annex I	The road safety impact assessment shall be carried out at the initial planning stage of the infrastructure project, before—(a) in the case of an infrastructure project coming within Part IV of the Act of 1993, submitting a scheme to An Bord Pleanála, pursuant to sections 47 and 49 of the Act of 1993, as amended by sections 9 and 11 of the Act of 2007, or (b) in any other case, submitting an application for consent for the infrastructure project under the planning and development Act 2000 (No. 30 of 2000) and Regulations made under Part XI of that Act

in the directive article. The N-gram and approximate string matching features of USM facilitate the identification of such cases of transposition.

4.2 Unsupervised text similarity models based on word and paragraph embeddings learned by shallow neural networks

In this section, we will investigate word and paragraph embedding models learned by shallow neural networks to identify the transposition of directives. The word embeddings obtained from Word2vec model have been utilized in many natural language processing applications. Word embeddings could be highly useful in a short text similarity task as they can be used to enrich the texts with external semantic knowledge learned from a large corpus (Kenter and De Rijke 2015). Enriching directive and NIM provisions with external legal vocabularies could also be useful to identify transpositions because European and national law may have different terminologies. However, the enrichment of directive and NIM provisions with EuroVoc thesaurus did not improve over TF–IDF and LSA similarity models to identify transpositions (Nanda et al. 2016). Therefore, in this section, we utilize word embeddings to develop semantic similarity models for identifying transpositions.

4.2.1 Word2Vec

Word2Vec is one of the most common model used to generate word embeddings from a large unlabelled corpus (Mikolov et al. 2013). Word2Vec is the general name for two models: continuous bag-of-words (CBOW) and skip-gram. Both models are composed of two layers: an embedding layer and a hidden layer. The aim of the network is to maximize the cross entropy between the softmax of the output vector[4] and the one-hot vector of the target word. CBOW is based on the idea of

[4] The output vector is computed by multiplying the embedding vector by the hidden layer.

Table 5 Most similar words for a given word as per Word2vec embeddings

Word	Nearest words
Board	Vessel, master, passenger, ship
Requirement	Condition, satisfy, meet, minimum
Notice	Document, notification, collate, file
Contract	Offer, agreement, entity, purchase

bag-of-words: given a word at position t, CBOW generates a vector averaging the embedding in the window $[t-d, t+d]$, where d is the size of the window. The averaged vector is then multiplied by the hidden layer to predict the next word. In the skip-gram model, given a word in position t, the surrounding words in a window of size $[t-d, t+d]$ are predicted. We used both skip-gram and CBOW models to generate word embeddings.

4.2.2 FastText

FastText (Bojanowski et al. 2016) is a word embedding model developed by Facebook. It offers the advantage of computing the word vectors of words which were not in the vocabulary of the training set. It substantially differs from Word2Vec in terms of the loss function and the way it computes the embedding of a word. Instead of using cross-entropy, it uses a binary logistic loss for randomly sampling negative words from the vocabulary. The embedding matrix contains character n-gram embeddings (of size 3, 4, 5 and 6). For a particular word, the n-gram embeddings that compose the word are retrieved from the matrix, summed together and multiplied by the hidden layer. The resulting vector is then passed to the loss function. Finally, the learned n-gram embedding is used to define the word embedding of all words inside the vocabulary. We utilize both CBOW and skip-gram models of fastText.

4.2.3 System description for text similarity models based on word and paragraph embeddings

We require a large amount of unlabelled legal text data to train a word embedding model. Word embeddings trained on a legal domain corpus have shown better performance on legal datasets than generic embeddings trained on Google News and Wikipedia (Cardellino et al. 2017). This is because the data used to train the embeddings is quite different from the test data (legal data) on which embeddings have to be evaluated. Therefore, we collected a corpus of European directives and national legislation to train the word embeddings. The European part consists of a multilingual parallel corpus of 4300 directives in English, French and Italian. The national part consists of the national legislation from 1960 to 2018 from Ireland, Luxembourg and Italy. The number of documents were 27,365; 14,365 and 16,233 in Ireland, Luxembourg and Italian legislation respectively. The embeddings were trained on this combined corpus of European directives and national legislation. The NLP pre-processing pipeline discussed in Sect. 3.2

was utilized to clean the corpus before training word embeddings. Table 5 presents the most similar words for four sample words as per the word embeddings trained on the English Directive-NIM corpus. The implementation was carried out in Python and utilized Gensim, NLTK, scikit-learn and Tensorflow libraries (Abadi et al. 2016; Bird and Loper 2004; Pedregosa et al. 2011; Řehůřek and Sojka 2010). The pre-trained word vectors, trained on a large corpus such as Wikipedia and Google News had a dimension of 300. Through our experiments we observed that embeddings of dimension 300 perform better on a large corpus where they are able to capture and represent more information. In case of a comparatively small legal corpus, a smaller embedding size is more suitable. We set embedding dimension to 128, number of negative samples to 16, context windows to 5, and the learning rate to 0.1 for word2vec. For fastText, we also chose the same number of embedding dimensions as 128 (so as to compare its performance with word2vec). We utilized the default hyperparameters for fastText: context window size: 5, number of negative samples: 5 and learning rate: 0.1.

4.2.4 Computing provision vector

In order to utilize word embeddings for text similarity of legal provisions, we need to compute provision vectors. This could be done in two ways: word-sum and word-average. In word-sum, the provision vector is generated by adding the vector of the words in the provision. Given a sequence of N words, the resulting vector e_{sum} is computed as follows:

$$e_{sum} = \sum_{i=1}^{N} e_i \tag{7}$$

where e_i is the embeddings of i-th word. In word-average, the sum of the word embeddings in a provision is divided by the provision length. The resulting average vector e_{avg} is computed as follows:

$$e_{avg} = \frac{\sum_{i=1}^{N} e_i}{N} \tag{8}$$

We also experiment with inverse document frequency (IDF) and word-sum. Since some words in a text are more relevant compared to others, we multiply each word embedding by the IDF of the word. The average-idf provision vector, e_{idf} is computed as follows:

$$e_{avg_{idf}} = \frac{\sum_{i=1}^{N} e_i \times idf_{w_i}}{N} \tag{9}$$

where idf_{w_i} is the IDF value of i-th word in the provision. The formula in Eq. 9 is very similar to TF–IDF, with the only exception that term-frequency is substituted by the embedding of the word.

4.2.5 Paragraph vector model

We also utilized paragraph vector, an unsupervised model which learns a fixed-length distributed vector representation for texts of variable length, such as sentences, paragraphs and documents (Le and Mikolov 2014). Paragraph vector model can be seen as an extension of word2vec. Word2vec involves predicting the target word given the context. The training data comprises (context, target word) pairs. The context may comprise not only the words but also other suitable features (for instance, part-of-speech tags of context words) which may help to predict the target word. Paragraph vector model adds a paragraph token to the context. This token represents the document or a paragraph as an additional context. This token also acts as the document or paragraph identifier. While training the word vectors, the paragraph vector is also trained. After the training is finished, the paragraph vector represents a distributed vector representation of the paragraph. The concatenation of word vectors with the paragraph vector is used to predict the next word. This model is called the Distributed Memory Model of Paragraph Vectors (PV-DM). Another variant of paragraph vector model is called Paragraph Vector without word ordering: Distributed bag of words (PV-DBOW). This method ignores the input context words which were used by the PV-DM method. It uses the paragraph vector along with an input word to predict other words in the paragraph. This model does not require to store word vectors and is thus much faster. Previous experiments have demonstrated that a paragraph vector obtained as a combination of PV-DM and PV-DBOW models achieves a better performance as compared to paragraph vectors obtained individually from each model (Le and Mikolov 2014). We also utilized a combination of PV-DM and PV-DBOW to develop provision vectors for the directive-NIM corpus. The paragraph vector model was trained on the combined unlabelled corpus of European directives and national legislation. We used the same dimension size of 128 as word2vec and fastText provision vectors.

Figure 2 displays the results of the word2vec model (for different provision vectors) for the multilingual corpus of directives and NIMs. We observe that the Luxembourg Directive-NIM corpus achieves the best precision, recall and F-score for different word2vec models. This result is coherent with the results of the similarity measures discussed in Sect. 4.1.5. The performance of both skip-gram and CBOW models of word2vec is comparable across the multilingual corpus. But the CBOW model slighlty outperforms the skip-gram model in terms of F-score for all three languages. The legal datasets of European directives and national legislation used in this paper to train word embeddings are quite small as compared Wikipedia or Google News datasets which are generally used to train the embeddings. The CBOW model smoothes most of the distributional information as it models the entire context as one observation (Abadi et al. 2016). As a result, CBOW achieves better performance than skip-gram in smaller datasets. The skip-gram model on the other hand considers each word-context pair as a new observation. Therefore, the

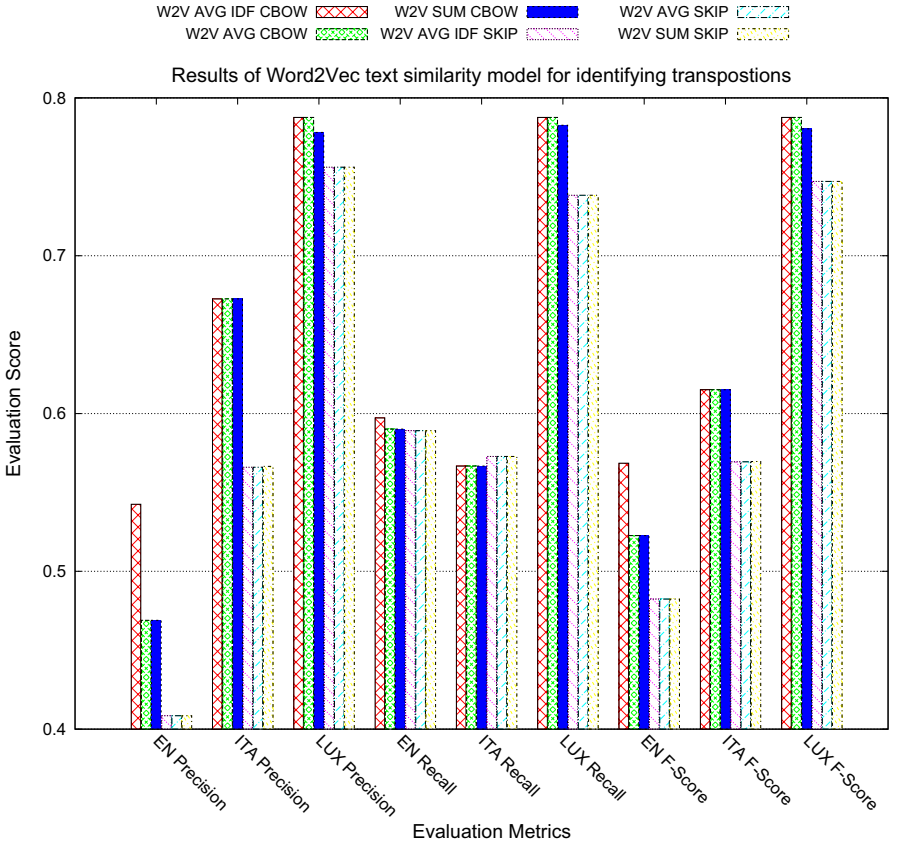

Fig. 2 Macro-average precision, recall and F-score values for skip-gram and CBOW Word2vec models

skip-gram model works better in case of a larger dataset as it provides a larger number of observations. The performance of different provision vector models for the CBOW model is comparable. The average-idf provision vector performs slightly better than other vectors in the English corpus. In French and Italian corpus, both average and average-idf vectors have the similar performance and slightly outperform the sum vector. Overall, we conclude that the average-idf had the best performance in the CBOW model. In case of the skip-gram model, all the provision vectors have similar performance. Figure 3 displays the results of the fastText model for the multilingual corpus of directives and NIMs. In this case also the Luxembourg Directive-NIM corpus achieves a higher F-score than English and Italian corpus. We also observe that the skip-gram model of fastText slightly outperforms the CBOW model. This is because the skip-gram model in word2vec predicts the context only from the vectors of words present in the training corpus. Whereas the skip-gram model of fastText utilizes the vectors of the word and also vectors of the n-grams comprising the word. The presence of n-grams results in achieving a better performance for syntactic tasks due to the addition of morphological

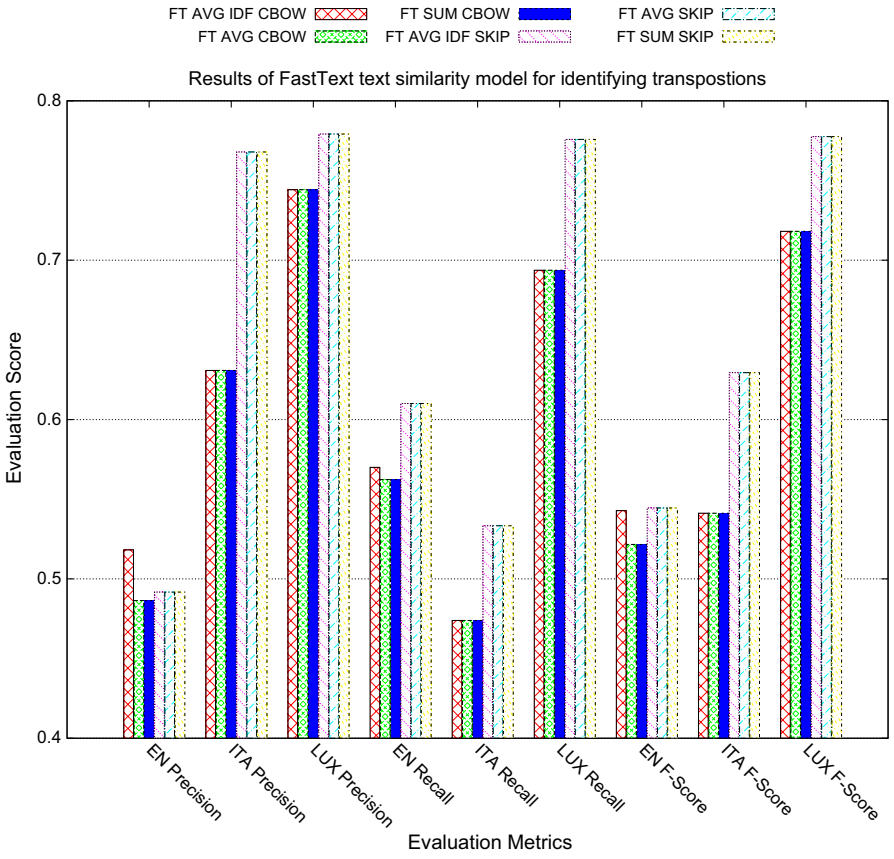

Fig. 3 Macro-average precision, recall and F-score values for skip-gram and CBOW models of FastText

information (Bojanowski et al. 2016). The performance of different provision vectors for both skip-gram and CBOW models is very similar. The average-idf vector has a slightly better performance than other vectors in case of the English corpus. We also evaluate the paragraph vector on the multilingual corpus of 43 directives and their corresponding NIMs. Figure 4 displays the results of the paragraph vector and the best performing provision vectors of word2vec (average-idf of the CBOW model) and fastText (average-idf for the skip-gram model) model. The results indicate that the paragraph-vector model outperforms both word2vec and fastText in terms of F-score. One advantage of using paragraph vectors is that they take into account the word order though in a small context (Le and Mikolov 2014). The provision vectors developed by the sum and average of word vectors lose the word order. Therefore, paragraph vector models show better performance to identify transpositions as compared to provision vector models of fastText and word2vec. We also present a two-dimensional visualization of provision vectors generated by fastText and latent semantic analysis (LSA) models as shown in Fig. 5 (fastText vectors are

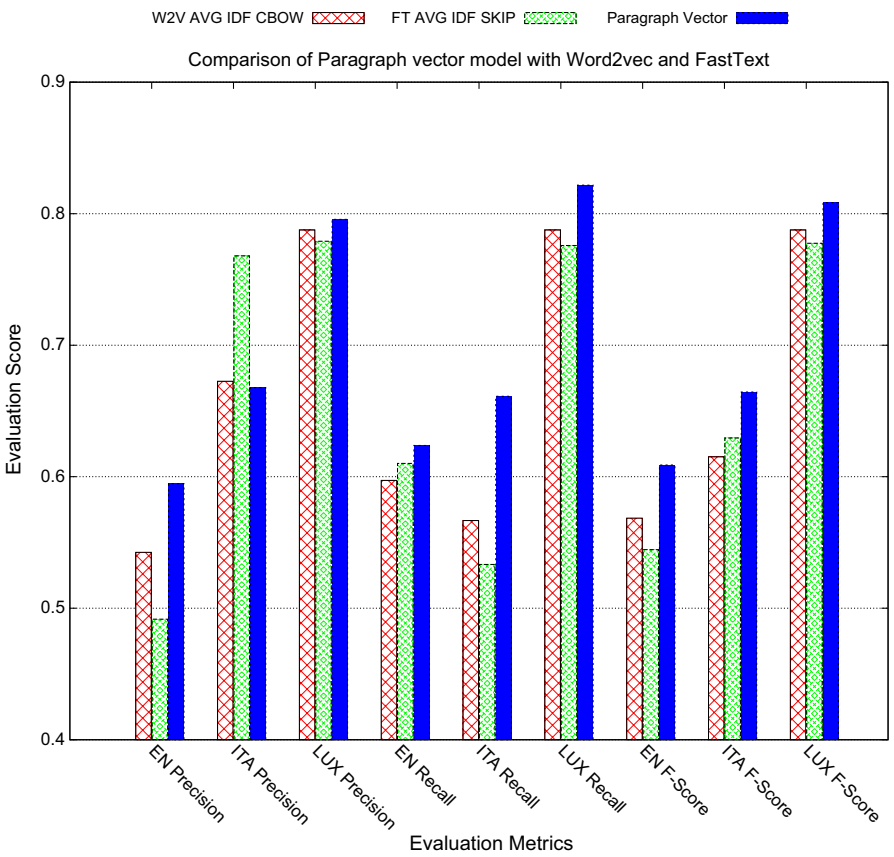

Fig. 4 Comparison of paragraph vector model with word2vec and fastText

represented by the top plot and LSA vectors are represented by the bottom plot). The visualization is generated by using t-Distributed Stochastic Neighbour Embedding (t-SNE) (Maaten and Hinton 2008). It is a dimensionality reduction algorithm which converts high-dimensional data into a low-dimensional (two or three-dimensions) space for visualization. In Fig. 5, the labels *A* and *P* represent the directive articles and NIM provisions respectively. We encircle some article and provision pairs in both plots which are very close to each other. We observe that the pairs encircled with blue colour (A10.1, P14.1), (A3, P2.1), (A9.1, P13) and (A2, P3.2) are clustered together in both fastText and LSA plots. These pairs of transposition were correctly identified by both fastText and LSA. In the LSA plot, we also encircle the pair (A7, P8.3), with light green colour, which was correctly identified by LSA but missed by fastText. In fastText plot, points A7 and P8.3 are far away and not clustered together. We observe that semantically similar provisions are mostly clustered together in the visualization. Moreover, we can also find correspondences between similar provisions from the same legislative document (for instance NIM provisions

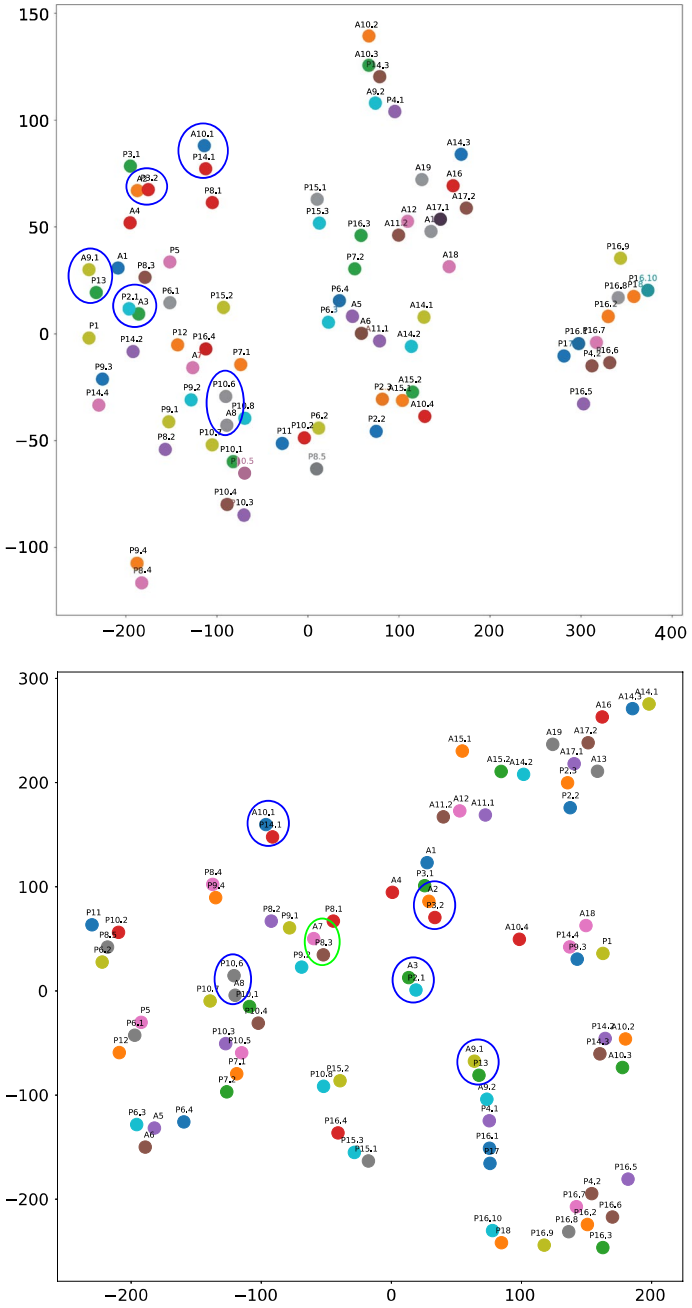

Fig. 5 Two-dimensional visualization of fastText (top plot) and LSA (bottom plot) provision vectors using t-SNE for directive CELEX 32001L0096 and Ireland NIM 72001L0096IRL_115977

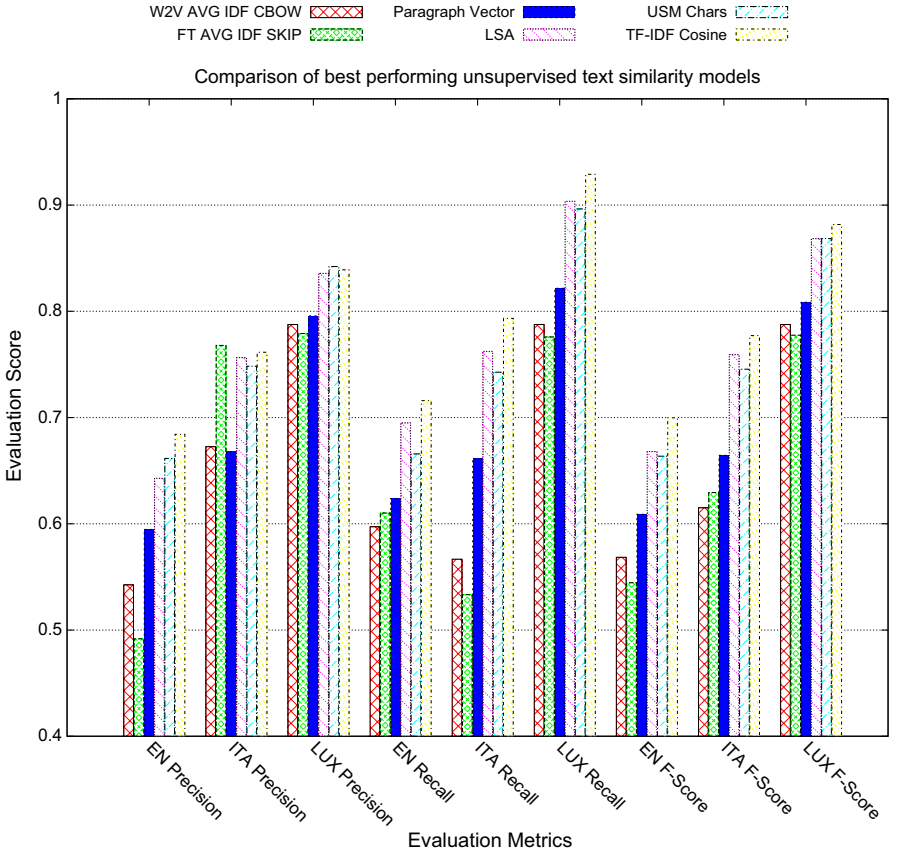

Fig. 6 Comparison of the best performing unsupervised text similarity models

P11, P10.2, P6.2 and P8.5 are clustered together in both plots). Figure 6 presents the results of the best performing unsupervised text similarity models (discussed in Sects. 4.1 and 4.2). We observe that TF–IDF cosine model had the best performance in terms of F-score for all three corpora. It was closely followed by LSA and USM_chars model. The lexical and semantic similarity methods outperform the word and paragraph embedding models. This is probably because a large number of transpositions can be identified by highlighting important terms using TF–IDF and modeling their relationships through LSA. The results of the embedding-based models are encouraging and probably with improvements in provision vector representation their performance may improve. There were some cases where they were able to identify the complex cases of transposition which were missed by the best performing methods. Table 6 presents an example of a transposition which was identified by paragraph vector and word2vec models but missed by all other methods such as TF–IDF cosine, USM, LSA, LDA and fastText. We observe that the NIM provision only partly implements the directive article. The second part of NIM provision talks

Table 6 Article 4.2 of directive (CELEX Number: 32009L0020) and its implementing NIM provision 4.3 from Ireland legislation (CELEX Number: 72009L0020IRL_188439)

Article 4.2 of directive	Provision 4.3 of Ireland NIM
Each member state shall require shipowners of ships flying a flag other than its own to have insurance in place when such ships enter a port under the member state's jurisdiction. This shall not prevent member states, if in conformity with international law, from requiring compliance with that obligation when such ships are operating in their territorial waters	The owner of a ship flying a flag other than that of the state—(a) shall have insurance in force in respect of the ship when it enters a port in the state, and (b) shall ensure that proof of such insurance in the form of a certificate or certificates referred to in regulation 5(2) is carried on board the ship

about the proof of insurance which is not mentioned in the directive article. The NIM also does not mention anything about compliance and conformity with international law as mentioned in the directive. The proximity of word vector pairs (trained on the legal corpus), such as "owners" and "shipowners", "in place" and "in force" facilitates the identification of this transposition.

5 Supervised machine learning techniques to identify transpositions

The techniques discussed in previous section are unsupervised as they utilize unlabelled dataset. In this section, we will investigate the application of supervised machine learning approaches to identify semantically similar legal provisions. The objective is to find the transposing NIM provisions for a particular article of the directive. We utilize the labelled training data from the gold standard for this purpose. If a directive article, A is transposed by a NIM provision, P then they are considered to be similar provisions (represented by "True" label). The provisions which are not similar are represented by the "False" label. The "False" label also implies that the NIM provision, P does not transpose the directive article, A. Therefore, this is a binary classification problem with two classes, "True" and "False". We select an equal number of "True" and "False" label pairs from the corpus to develop a balanced dataset. Both "True" and "False" label pairs were selected from the intersection set of both annotators. Table 7 shows the format of the dataset used for this

Table 7 Dataset format for supervised classification of provisions

Directive article	NIM provision	Transposition
A1	P1	True
A2	P2	True
A3	P3	False
A4	P4	True
...	...	...
A101	P43	Classifier predicts? True/False

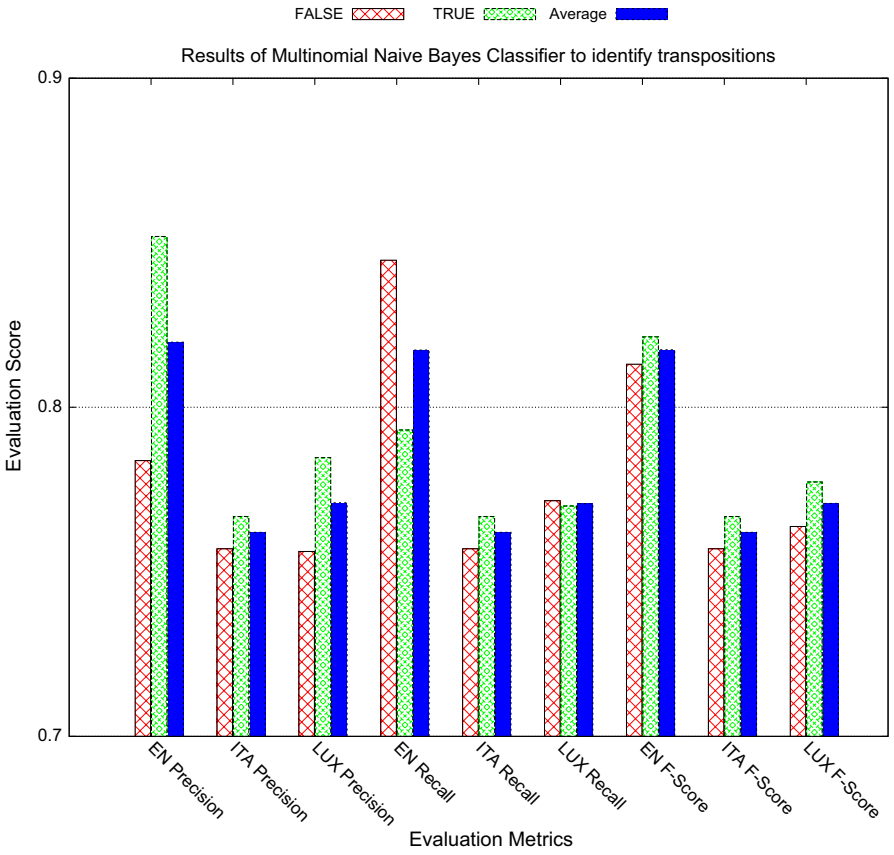

Fig. 7 Results of Multinomial Naive Bayes to identify transpositions

classification task. The directive articles A and NIM provisions P represent the text of each article and provision respectively. The directive articles and NIM provisions are first passed through the NLP pre-processing pipeline as discussed in Sect. 3.2. We utilize TF–IDF vectors for feature extraction. The dataset was divided into 80% training and 20% test set. We utilized the Mutltinomial Naive Bayes classifier as the baseline model. Figure 7 presents the results of the Multinomial Naive Bayes classifier to identify both similar ("True") and not similar ("False") provisions. The overall precision (represented by Average) for both classes is computed as

$$weighted\_precision = \frac{P_T \times |T| + P_F \times |F|}{|T| + |F|} \tag{10}$$

where P_T and P_F are the precision values for class True and False, and $|T|$ and $|F|$ are the number of instances in True and False class. The weighted recall is also computed in a similar way as per Eq. 6, but, by using recall values from both classes. We observe that the English Directive-NIM corpus achieves the highest precision,

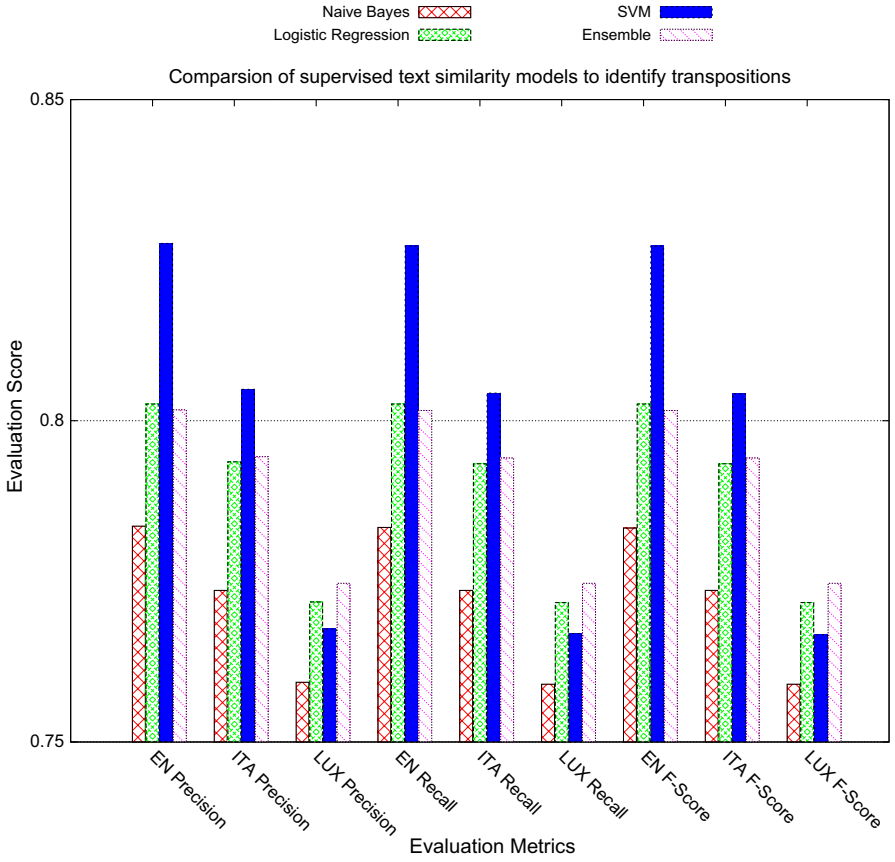

Fig. 8 Comparison of different machine learning classifiers over tenfolds cross-validation

recall and F-score. The results indicate that Naive Bayes classifier is quite effective in differentiating both True and False class labels across all the three legislations. We further evaluated logistic regression, support vector machines (SVM), multinomial Naive Bayes and an ensemble classifier over tenfolds cross-validation using TF–IDF features. The ensemble classifier is a voting classifier which is used to combine conceptually different machine learning classifiers (Pedregosa et al. 2011). A majority vote is used to decide the predicted class label. Figure 8 presents the results (weighted average values of precision, recall and F-score over both class labels) of different classifiers on the multilingual corpus. The results indicate that SVM classifier has the best performance in Italian and English legislation. This result is consistent with previous findings where SVM has been shown to outperform other classifiers for text classification (Joachims 1998; Boella et al. 2013). In case of Luxembourg legislation, the ensemble classifier outperforms other classifiers. The F-score values (for Luxembourg corpus) of logistic regression and SVM are comparable and we observe the benefit of using the ensemble classifier in this

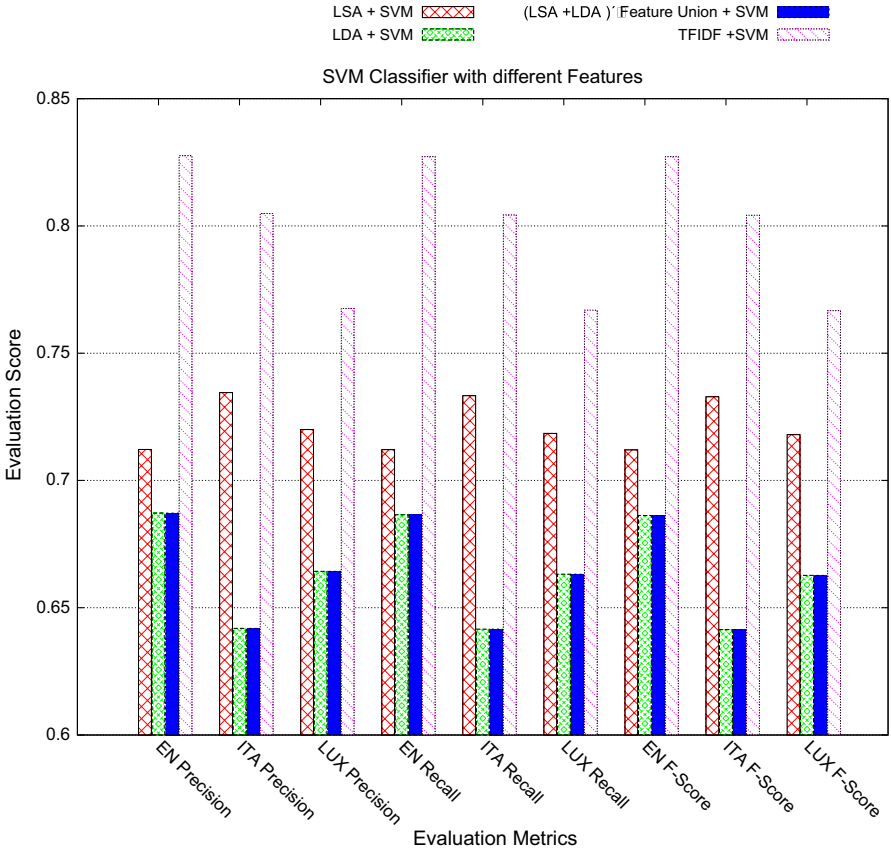

Fig. 9 The performance of SVM classifier with different features

case. We utilized the SVM classifier to experiment with different features due to its overall good performance over the multilingual corpus. We used LSA and LDA vectors as features for the classifier. A feature union of LSA and LDA features was also used. The feature vectors from LSA and LDA transforms are extracted individually and are then concatenated into a single transform. Figure 9 presents the results of the SVM classifier with different features for tenfolds cross-validation. The results indicate that TF−IDF + SVM outperforms LSA + SVM, LDA + SVM and (LSA + LDA) Feature Union + SVM. This also corroborates the results of the unsupervised methods where TF–IDF Cosine had the best performance.

6 Conclusion and future work

This paper presented a thorough investigation of both unsupervised and supervised text similarity models to identify the transpositions of European directives. The models were evaluated on a multilingual corpus of 43 directives and their corresponding

NIMs from Ireland, Luxembourg and Italy. Our results indicate that the lexical and semantic unsupervised methods had a better performance than word and paragraph embedding models. However, the word and paragraph embedding models were successful in identifying certain types of transpositions which were missed by other methods. The SVM classifier showed promising results with different features set. The TF−IDF + SVM model had the best performance among the supervised text similarity models. The best performance in identifying transpositions was achieved by utilizing the TF–IDF features for both supervised and unsupervised methods. The best performing unsupervised similarity measure, TF–IDF Cosine had macro-average F-Score of 0.8817, 0.7771 and 0.6997 for the 43 directive-NIM corpus of Luxembourg, Italy and Ireland respectively. These results indicate that such legal information retrieval systems can be used to semi-automate the manual task of identifying transpositions in different Member States. In the future work, we intend to utilize legal concept recognition systems (Nanda et al. 2017b), word-sense disambiguation, as well as shallow semantic representations based on flat reification-based approaches (Robaldo 2010, 2011; Robaldo and Sun 2017), to develop text similarity models for identifying transpositions. It would also be interesting to study the influence of other linguistic and legal features for the supervised classifiers, as well as devising *hybrid* (rule-based and statistical) approaches by integrating rule-based systems such as Robaldo et al. (2011).

Acknowledgements Research presented in this paper is conducted as a Ph.D. research at the University of Turin, within the Erasmus Mundus Joint International Doctoral (Ph.D.) programme in Law, Science and Technology. This work has been partially supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skodowska-Curie Grant agreement no. 690974 for the project "MIREL: MIning and REasoning with Legal texts".

References

Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M et al (2016) Tensorflow: a system for large-scale machine learning. In: OSDI, vol 16, pp 265–283

Ajani G, Boella G, Di Caro L, Robaldo L, Humphreys L, Praduroux S, Rossi P, Violato A (2017) The European legal taxonomy syllabus: a multi-lingual, multi-level ontology framework to untangle the web of European legal terminology. Appl Ontol 2(4):325–375

Aletras N, Tsarapatsanis D, Preoţiuc-Pietro D, Lampos V (2016) Predicting judicial decisions of the European court of human rights: a natural language processing perspective. PeerJ Comput Sci 2:e93

Bergamaschi S, Po L (2014) Comparing lda and lsa topic models for content-based movie recommendation systems. In: International conference on web information systems and technologies. Springer, pp 247–263

Bird S, Loper E (2004) Nltk: the natural language toolkit. In: Proceedings of the ACL 2004 on interactive poster and demonstration sessions. Association for Computational Linguistics, p 31

Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3(Jan):993–1022

Boella G, Di Caro L, Humphreys L, Robaldo L, van der Torre L (2012) Nlp challenges for eunomos, a tool to build and manage legal knowledge. In: Language resources and evaluation (LREC). pp 3672–3678

Boella G, Di Caro L, Robaldo L (2013) Semantic relation extraction from legislative text using generalized syntactic dependencies and support vector machines. Springer, Berlin, pp 218–225

Boella G, Di Caro L, Humphreys L, Robaldo L, Rossi R, van der Torre L (2016) Eunomos, a legal document and knowledge management system for the web to provide relevant, reliable and up-to-date information on the law. Artif Intell Law 24:245–283

Bojanowski P, Grave E, Joulin A, Mikolov T (2016) Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606

Cardellino C, Teruel M, Alemany LA, Villata S (2017) A low-cost, high-coverage legal named entity recognizer, classifier and linker. In: Proceedings of the 16th edition of the international conference on artificial intelligence and law. ACM, pp 9–18

Ciavarini Azzi G (2000) The slow march of european legislation: the implementation of directives. In: European integration after Amsterdam: institutional dynamics and prospects for democracy

Cosma G, Joy M (2012) An approach to source-code plagiarism detection and investigation using latent semantic analysis. IEEE Trans Comput 61(3):379–394

Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R (1990) Indexing by latent semantic analysis. J Am Soc Inf Sci 41(6):391

Eliantonio M, Ballesteros M, Rostane M, Petrovic D (2013) Tools for ensuring implementation and application of eu law and evaluation of their effectiveness. Technical reports on European Parliament

Golub GH, Reinsch C (1970) Singular value decomposition and least squares solutions. Numer Math 14(5):403–420

Hartung J, Knapp G, Sinha B (2011) Statistical meta-analysis with applications, vol 738. Wiley, Hoboken

Hong L, Davison BD (2010) Empirical study of topic modeling in twitter. In: Proceedings of the first workshop on social media analytics. ACM, pp 80–88

Humphreys L, Santos C, Di Caro L, Boella G, Van Der Torre L, Robaldo L (2015) Mapping recitals to normative provisions in eu legislation to assist legal interpretation. In: JURIX. pp 41–49

Joachims T (1998) Text categorization with support vector machines: learning with many relevant features. In: European conference on machine learning. Springer, pp 137–142

Kenter T, De Rijke M (2015) Short text similarity with word embeddings. In: Proceedings of the 24th ACM international on conference on information and knowledge management. ACM, pp 1411–1420

Le Q, Mikolov T (2014) Distributed representations of sentences and documents. In: International conference on machine learning. pp 1188–1196

Maaten LVD, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9(Nov):2579–2605

Magerman T, Van Looy B, Song X (2010) Exploring the feasibility and accuracy of latent semantic analysis based text mining techniques to detect similarity between patent documents and scientific publications. Scientometrics 82(2):289–306

Mandal A, Chaki R, Saha S, Ghosh K, Pal A, Ghosh S (2017) Measuring similarity among legal court case documents. In: Proceedings of the 10th annual ACM India compute conference, Compute '17. ACM, New York, pp 1–9

McHugh ML (2012) Interrater reliability: the kappa statistic. Biochem Med 22(3):276–282

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

Nanda R, Di Caro L, Boella G (2016) A text similarity approach for automated transposition detection of European union directives. In: 29th International conference on legal knowledge and information systems, JURIX 2016, vol 294. IOS Press, pp 143–148

Nanda R, Di Caro L, Boella G, Konstantinov H, Tyankov T, Traykov D, Hristov H, Costamagna F, Humphreys L, Robaldo L, et al (2017) A unifying similarity measure for automated identification of national implementations of European union directives. In: Proceedings of the 16th edition of the international conference on articial intelligence and law. ACM, pp 149–158

Nanda R, Siragusa G, Caro LD, Theobald M, Boella G, Robaldo L, Costamagna F (2017) Concept recognition in European and national law. In: Legal knowledge and information systems—JURIX 2017: the thirtieth annual conference, Luxembourg, 13–15 December 2017, pp 193–198

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830

Řehůřek R, Sojka P (2010) Software framework for topic modelling with Large Corpora. In: Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks, ELRA, Valletta, Malta, pp 45–50. http://is.muni.cz/publication/884893/en

Robaldo L (2010) Interpretation and inference with maximal referential terms. J Comput Syst Sci 76(5):373–388

Robaldo L (2011) Distributivity, collectivity, and cumulativity in terms of (in)dependence and maximality. J Log Lang Inf 20(2):233–271
Robaldo L, Sun X (2017) Reified input/output logic: combining input/output logic and reification to represent norms coming from existing legislation. J Log Comput 27:2471–2503
Robaldo L, Caselli T, Russo I, Grella M (2011) From Italian text to timeml document via dependency parsing. In: Computational linguistics and intelligent text processing—12th international conference, CICLing 2011, Tokyo, Japan, 2011, pp 177–187
Sparck Jones K (1972) A statistical interpretation of term specificity and its application in retrieval. J Doc 28(1):11–21

Affiliations

Rohan Nanda[1] · Giovanni Siragusa[1] · Luigi Di Caro[1] · Guido Boella[1] · Lorenzo Grossio[2] · Marco Gerbaudo[2] · Francesco Costamagna[2]

Giovanni Siragusa
siragusa@di.unito.it

Luigi Di Caro
dicaro@di.unito.it

Guido Boella
guido@di.unito.it

Lorenzo Grossio
lorenzo.grossio@edu.unito.it

Marco Gerbaudo
marco.gerbaudo@edu.unito.it

Francesco Costamagna
francesco.costamagna@unito.it

[1] Department of Computer Science, University of Turin, Corso Svizzera, 185, 10149 Turin, Italy

[2] Department of Law, University of Turin, Lungo Dora Siena 100/A, 10153 Turin, Italy