algorithms

MDPI

Article

Ensemble and Deep Learning for Language-Independent Automatic Selection of Parallel Data

Despoina Mouratidis * and Katia Lida Kermanidis

Department of Informatics, Ionian University, 491 00 Kerkira, Greece; kerman@ionio.gr
* Correspondence: c12mour@ionio.gr; Tel.: +30-266-1087756

Received: 31 October 2018; Accepted: 15 January 2019; Published: 18 January 2019

Abstract: Machine translation is used in many applications in everyday life. Due to the increase of translated documents that need to be organized as useful or not (for building a translation model), the automated categorization of texts (classification), is a popular research field of machine learning. This kind of information can be quite helpful for machine translation. Our parallel corpora (English-Greek and English-Italian) are based on educational data, which are quite difficult to translate. We apply two state of the art architectures, Random Forest (RF) and Deeplearnig4j (DL4J), to our data (which constitute three translation outputs). To our knowledge, this is the first time that deep learning architectures are applied to the automatic selection of parallel data. We also propose new string-based features that seem to be effective for the classifier, and we investigate whether an attribute selection method could be used for better classification accuracy. Experimental results indicate an increase of up to 4% (compared to our previous work) using RF and rather satisfactory results using DL4J.

Keywords: machine learning; deep learning; education data; data selection; machine translation; DL4J deep learning architecture; random forest

1. Introduction

Machine learning technology is used in many aspects of modern life. Machine learning applications are used to identify objects in images, transcribe speech into text, select relevant results in a searching task, machine translation and others. Lately, machine learning has been employing new architectures, namely deep learning techniques. Deep neural networks have proven successful in various scientific fields. Additionally, recent work showed that neural networks can be successfully used in several tasks in natural language processing (NLP) [1,2]. Traditionally machine translation models were statistically based (SMT) [3]. Recent approaches show that models based on neural machine translation (NMT) are more effective [4]. The last ones reach, in many cases, better machine translation quality, because they can use only the words that have information relevant to the target word, without wasting time to calculate useless probabilities [5]. Nowadays, translation platforms give the opportunity to edit the translation output, in order for experts to improve it [6], this process is quite helpful, and generates a lot of data that must be classified as useless or not for the translation model.

For classification, a state of the art algorithm family is decision trees [7]. On the other hand, deep learning methods show promising results [8]. In some cases, deep learning achieves better results than decision trees, but requires more training time [9]. In this paper, we used two classifiers, one from the decision trees family, i.e., Random Forest, and one deep learning architecture, Deeplearning4j.

Contribution and Novelties

The current work constitutes a continuation of a preliminary study [10]. Compared to this earlier work, the present approach includes the following novelties:

- the inclusion of 14 additional linguistic features.
- the use of a Deep Learning architecture (DL4J) for classification.
- the exploration of different validation options (k-fold Cross Validation (CV) and Percentage split).
- the application of the methodology to an additional language pair (English-Italian), for enabling a comparative inter-linguistic analysis of results.
- the application of feature selection methods.

To the author's knowledge, this is a first time that deep learning methods are used for a classification task on this kind of data.

The rest of the paper is organized as follows. Section 2 presents the related work of the scientific area. Section 3 describes the materials and methods, the data sets (corpora), the tools and our process (the various feature sets, the annotation process etc.). Section 4 describes more experimental details and the results of the classification process. Finally, Sections 5 and 6 present our conclusions and directions for future research.

2. Related Work

Recently, sequence to sequence techniques have been applied to various NLP tasks with promising results. In machine translation, it can be used as a part of the SMT system [11–13]. The performance of a statistical machine translation system is found to be improved by using the Encoder–Decoder models, which are a generic deep-learning approach to sequence-to-sequence tasks [5].

There are two basic architectures used, the first one is a model of two Recurrent Neural Networks (RNNs) [12]. The first RNN encodes a sequence of symbols into a fixed-length vector representation, while the other RNN decodes the representation into another sequence of symbols. The network is trained to maximize the conditional probability of a target sequence given a source sequence, and learns the semantic and syntactic representation of the corpus [12], which gives the opportunity for learning embeddings from the language as well. The embeddings provide rich linguistic information on a language, so a model trained with embeddings is able to build strong representations of the language [14]. The second is an Encoder-Decoder model with two Long Short-Term Memory (LSTMs) layers [15]. The LSTMs architecture is one of the most important deep architectures for natural language processing (NLP) [16,17]. Usually, it is followed by a dense/fully connected layer (every input connected to every output), and the output activation layer using the Softmax function it converts every vector into class probabilities [8,18]. Following the principles of RNNs, LSTM networks are generic in the sense that given adequate network units, they allow for any conventional computation. In contrast to RNNs, LSTMs are more suited for learning from experience for classification, as well as for processing and predicting time-series. [15] showed that the dependencies between the source and the target sentences made the learning problem simpler. These architectures are powerful because their LSTMs can examine the output structure.

Many different learning algorithms are used for classification purposes, such as Naïve Bayes, the implementation of the C.4.5 algorithm J48, Support Vector Machine (SVM) and Random Forest. Naïve Bayes as a classifier, assumes that the values of feature are independent to any other feature [19,20]. It also uses a decision rule (maximum a posteriori) in order to select the probable hypothesis [19]. J48 builds decision trees with the help of information entropy and it is only handling numeric data [21]. SVMs are based on statistical learning theory (SLT) and they use support vectors to represent decision boundaries from the training data. Phyu, et al. [22] uses SVMs for text classification. The data (parallel data) consists of one morphologically rich language (Croatian) and one that is not so rich in morphology (English). SVMs are designed for binary classification problems [19,23].

The Random Forest learning classifier is used in many classification problems. Random Forest classifier is an ensemble of decision trees (it consists of a combination of tree classifiers). Decision tree classifier is run iteratively on a random subsample of the feature space, generating multiple trees, the final decision is made based on majority voting. Many works have showed satisfactory results in image classification [24–26], remote sensing [27] and others. In text classification, RF outperforms other popular text classification methods, such as SVM, NB and KNN. A change in classical RF (for example a change in the feature selection method) can generate better classification accuracy [28]. That is because RF uses only a portion of the input features for each split which makes it computationally lighter [29]. Deep learning is the state of the art method in many tasks [30,31]. Deep Learning (DL) classification methods differ from statistical classification methods (RF) in that DL requires more training data, and more time for training as well. Furthermore, DL algorithms, require more parameters to be defined, compared to traditional classifiers such as RF. Deep Learning classification has showed promising results when facing complex tasks, such as speech recognition [32], image classification [33,34], natural language processing [2], question answering (phonetic classification) [18], language translation [15] and others [33].

There is limited work on using sequence to sequence comparison for identifying useful data selection feedback for machine translation. [35] introduced an analysis of an annotated corpus based on automatic translation and user-provided translation request-corrections gathered through an online machine translation system. Barrón-Cedeño, et al. [36] proposed an extension of the corpus described in [35]. They calculate new features and they try different configurations of SVM as well. Both papers showed that the translation quality of a phrase-based SMT system can be improved by using human correction feedback. Unlike previous work, the present study uses the inclusion of 14 additional linguistic features and it explores different validation options (k-fold Cross Validation (CV)-Percentage split). For enabling a comparative inter-linguistic analysis of results, an additional language pair (English-Italian) is used. To the author's knowledge, this is the first time that deep learning methods are used for a classification task on this kind of data.

3. Materials and Methods

Here we describe in detail the data sets, tools and classification process used in our experiments.

3.1. Corpora

The corpora that we used in our experiment are from the TraMOOC project [37]. The corpora consists of educational data, lecture subtitle transcriptions etc., with unorthodox syntax, ungrammaticalities etc. The English–Greek corpus is described in detail by Mouratidis [10]. In the present work an additional parallel English-Italian corpus is also employed, taken from the same project. The source of the English-Italian corpus consists of 2745 segments in English (Src). For each of these segments, three translation outputs in Italian were provided, generated by three prototypes (Trans1, Trans2, Trans3). Also, a professional translation is provided (Ref), and used as a reference for each language. The models that we used are: Trans1 (based on the phrase-based SMT toolkit Moses [26]), Trans2 and Trans3 (based on the NMT Nematus toolkit [38]). The models (Trans1, 2, 3) are trained in both in- and out- of domain data. Trans3 is trained on additional in-domain data provided via crowdsourcing, and also includes layer normalization and improved domain adaptation. Out-of-domain data included widely known corpora e.g., Europal, JRC-Acquis, OPUS, WMT News corpora etc. In-domain data included data from TED, the QED corpus, Coursera, etc. [39]. It was also necessary to perform data pre-processing, for example, removing special symbols (@, /, #), including alignments corrections, each segment matching to the other (Src-Trans1-2-3-Ref).

3.2. Annotation

Two Italian language experts annotated each text segment with A, B or C if the Trans 1, 2 or 3 output is closer in meaning to the Ref translation respectively. We observe similar annotation results with the English-Greek corpus [10], namely class C presents the highest frequency value. Additionally, in English-Italian corpus, class A got higher value than in English-Greek corpus. More specifically, the frequency distribution among the classes is as follows: class A: 22%, class B: 32% and class C: 46%. Again it seems that the NMT models perform better than the SMT models. Below, in Table 1, you can see 5 examples of segments. Their first part is the English source text, followed by the three Italian machine translation outputs (Trans1, 2, 3), and the final part is the Italian reference translation.

Table 1. Segments example of Source, Trans1, 2, 3 and Reference.

ID	Source	Trans1	Trans2	Trans3	Reference
1	I think it's still a fortune don't be in the same conditions he is'	Credo sia ancora una fortuna non essere nelle stesse condizioni ?	Credo che sia ancora una fortuna non trovarsi nelle stesse condizioni.	Penso che sia ancora una fortuna non essere nelle stesse condizioni che fa lui.	Penso che sia ancora una fortuna non essere nelle sue stesse condizioni.
2	By the end of the video, I realized I take about 40% critically reasoned decisions and executed but 60% of the decisions go emotionally reasoned or executed.	Alla fine del video, ho capito che mi prendo il 40% criticamente decisioni motivate e giustiziato ma il 60% delle decisioni andare emotivamente motivata o giustiziati.	Entro la fine del video, ho capito che prendo circa il 40% di decisioni critiche e giustiziate, ma il 60% delle decisioni prese emotivamente motivate o eseguite.	Alla fine del video, ho capito che prendo circa il 40% decisioni motivate e eseguite, ma il 60% delle decisioni vanno emotivamente motivate o eseguite.	Alla fine del video, mi sono reso conto che circa il 40% sono decisioni ragionate criticamente e messe in pratica, ma il 60% delle decisioni sono ragionate emozionalmente o messe in pratica.
3	Its a platform you learn different human behaviours, social interactive and changes.	La sua piattaforma impara diversi comportamenti umani, i cambiamenti sociali e interattivi.	La sua piattaforma impara diversi comportamenti umani, interattivi e cambiamenti sociali.	? una piattaforma che impari comportamenti umani diversi, social interattivi e cambiamenti.	È una piattaforma dove si conoscono diversi comportamenti umani, interazioni sociali e cambiamenti.
4	The whole world is getting a platform to know each other.	Il mondo intero sta diventando una piattaforma a conoscerci.	Tutto il mondo riceve una piattaforma per conoscersi l'un l'altro.	Tutto il mondo sta ottenendo una piattaforma per conoscerci.	Il mondo intero diventa una piattaforma per conoscersi.
5	Hello, Im studing fashion design and my aim it's to become a sustainable and ethical fashion designer.	Ciao, Im studing fashion design e il mio obiettivo ? di diventare una stilista sostenibile ed etico.	Ciao, sto studiando la moda di moda e il mio obiettivo ? diventare un designer di moda sostenibile e etico.	Ciao, Im studing design design e il mio obiettivo ? diventare un designer di moda sostenibile ed etico.	Salve, studio creazioni di moda e il mio obiettivo è di diventare uno stilista etico e sostenibile.

We point out that, in the corpus English-Italian, in a lot of segments the translations of Trans 2 and Trans 3 (and sometimes Trans 1 too) are almost identical. We give three typical examples in Table 2.

Table 2. Examples of similar segments.

ID	Source	Trans 1	Trans 2	Trans 3	Reference
6	Nothing yet but they did say there was a LOT of students in this course.	Ancora niente, ma hanno detto che c'era un sacco di studenti in questo corso.	Ancora niente, ma hanno detto che c'era una LOT di studenti in questo corso.	Ancora niente, ma hanno detto che c'era un LOT di studenti in questo corso.	Ancora niente ma hanno detto che c'erano un SACCO di studenti in questo corso.
7	I thought it would be nice if there was one lesson every day instead of giving us a sack full of lessons on one day itself.	Pensavo che sarebbe bello se ci fosse una lezione ogni giorno invece di darci un sacchetto pieno di lezioni di un giorno.	Ho pensato che sarebbe stato bello se ci fosse una lezione ogni giorno invece di darci un sacco di lezioni in un solo giorno.	Ho pensato che sarebbe bello se ci fosse una lezione ogni giorno invece di darci un sacco di lezioni su un giorno.	Ho pensato che sarebbe bello se ci fosse una lezione al giorno invece del sacco pieno di lezioni che ci date in un giorno.

Table 2. *Cont.*

ID	Source	Trans 1	Trans 2	Trans 3	Reference
8	I believe that the creators give us enough time and it's also fair because you want to know your results and don't want to wait until the mooc is finished for it.	Credo che i creatori darci tempo ed ? anche giusto perch? vuoi sapere i vostri risultati e non voglio aspettare la mooc ? finito.	Credo che i creatori ci diano abbastanza tempo ed ? anche giusto perch? volete conoscere i vostri risultati e non volete aspettare che il mooc sia finito.	Credo che i creatori ci diano abbastanza tempo ed ? anche giusto perch? volete conoscere i vostri risultati e non vogliamo aspettare che il mooc sia finito.	Credo che gli autori ci abbiano dato abbastanza tempo, oltre a essere scorretto perché pretendi di conoscere i risultati prima del termine del mooc.

ID: 1 We have chosen Trans 3 for two reasons:

i. The translation of *think*: *Penso*, is the same as in the Reference.
ii. Even though it is not the best translation, Trans3 is the only model that translated the phrase *he is*.

ID: 2 We have chosen Trans 3 for three reasons:

i. The translation of *By the end*: *Alla fine del*, is the same as in the Reference.
ii. Trans 3 is the only model that translated correctly the past participle *executed*: *eseguite*.
iii. Trans 3 is the only model that translated the verb *go*: *vanno.* Although, as Trans 1 and Trans 2 did, Trans 3 also translated *emotionally reasoned* by a pleonasm: *emotivamente motivate*.

ID: 3 We have chosen Trans 3 for two reasons:

i. Trans 3 is the model that did not erroneously translate *Its* as a possessive adjective, as Trans 1 and Trans 2 did (*La sua*). In the segment *Its* is not a possessive adjective but the third person from of the Present Tense of the verb *to be* (*It's*). We consider that Trans 3 recognized that *Its* is the verb *to be* but it did not put *È*, as Trans 3 "always" puts a question mark to all accented vowels, as Trans 1 and Trans 2 also do.
ii. Trans 3 recognized and translated (*che impari*) the dependence relation between the main clause (*It's a platform*) and the subordinate clause (*you learn*).

On the other hand, Trans 3 did not put first (and correctly) the word *diversi*, as Trans 1 and Trans 2 did.

ID: 4 We have chosen Trans 1 for three reasons:

i. Trans 1's translation of *The whole world*: *Il mondo intero* is the best translation, enclosing the *whole* meaning of the word and is the same as the Reference's. Trans 2 and Trans 3' translation: *Tutto* is not so strong for accurately translating the word *whole*.
ii. Trans1 is the only model that translated the multi-semantic verb *to get* correctly (*diventare*).
iii. Trans1 translated the Present Continuous Tense correctly: *is getting* by *sta diventando*, giving thereby the sense of the present, of the action and of its duration. Trans3 did the same, but with the wrong verb.

On the other hand, Trans1 did not translate *to* by *per* to give the sense of this target preposition.

ID: 5 We have chosen Trans 2 for one main reason:

Trans 2 is the only model that correctly translated the verb *I'm studing*, erroneously written in the *Source* (without the apostrophe). We suppose that because of this error, Trans 1 and Trans 3 did not translate it. On the other hand, Trans 2 did not translate the words *fashion designer*, as only Trans 1 did. The three Trans translated in a different and erroneous way the words *fashion design*: Trans 1 did not translate them at all, Trans 2 translated both of them in Italian, but using the same word (*moda*), and Trans 3 did not translate the word *design* but translated the word *fashion* in English.

3.3. Features

As we mentioned in our previous work, we have a classification problem with three output values, which are the classes A, B and C. Every segment was represented as a tuple (Src, Trans 1-2-3, Ref). Each tuple was modeled as a feature-value vector, and all its dimensions are string-similarity, language independent features. Our preliminary approach (involving 82 features and Random Forest classification) [10] is initially applied to the English–Italian corpus. A part of our feature set was based on the work by Barrón-Cedeño et al. [27] and Pighin et al. [35]. The features were grouped into three categories:

1. Basic-Simple Features (e.g., lengths in words/characters, some ratios, distances like Levenshtein [40], vocabulary containment [41], etc).
2. Noise-Based Features (nominal features (True-False) e.g., identifying long segments, in order to check the translation ability).
3. Similarity-Based Features (using a length factor [42]).

In the present work, four novel features groups, that belong to the first category have been employed (increasing thereby the feature dimensions from 81 to 96).

- the length in characters of Src-Trans 1-Trans 2-Trans 3-Ref.
- the Jaccard's Index (for Ref-Trans 1, Ref-Trans 2, Ref-Trans 3), which is an efficient metric for string comparison. It is calculated at a token level, and compares two segments. It is defined as the ratio of the absolute value of the intersection of the words between the two segments to the absolute value of the union of the two segments. [6].
- the Dice distance [6], which is related to Jaccard's Index.

$$\text{Dice distance} = 2\text{xaccard Index}/(1 + \text{Jaccard Index}), \qquad (1)$$

- the Suffix feature. We observed that we have better translation when the translation output takes into account and does not alter the gender and grammatical number (singular or plural) of declinable words. This feature is calculated as follows:
  - ○ Step 1: The last character of every word were extracted from the Ref and the Trans 1, 2, 3 segments.
  - ○ Step 2: The number of common last character for segments Ref-Trans 1, Ref-Trans 2 and Ref-Trans 3 was calculated.
  - ○ Step 3: The ratio between the number and the number calculated in Step 2 and the total number of last characters.

All feature values were calculated using MATLAB.

3.4. Classifiers

Firstly, we chose to use for this kind of data, the Random Forest as a classifier. Random Forest is an ensemble algorithm that reaches good accuracy levels, and prevents overfitting, by creating random subsets of the features and building smaller trees [7]. We set the number of trees to be constructed to 65 (the number of iterations). 20 randomly chosen attributes were used for constructing each tree. We employed 10 fold CV for testing.

In our experiments we used the DL4J [43] deep learning library with the Weka [44] framework as backend. DL4J is designed to handle large corpora. After experimentation, we chose the following deep neural network architecture:

- Number of iterations: 3150/3340 (EN-GR/EN-IT)
- Number of layers: 3
- Type of layers: LSTM, Dense, Output layer
- Size of layers: First—Input 96 Output 96; Second—Input 95 Output 95; Third—Input 95 Output 3;
- Output layer: Activation Softmax
- Learning rate: 0.001
- Weight Initialization: XAVIER
- Activation Function: Activation RELU
- Lossfunction: LossMCXENT

We generally used the default Weka settings. According to DL4J's documentation an iteration is one update of the neural net model's parameters. Weka uses by default the number of instances to be the iterations.

4. Results

In this section, we will show more details about our experiments and their results. Most of our features were numerical and were normalized by using the Feature scaling method. Our experiment was run using the Weka machine learning workbench [44] for training and testing. For the classifier's evaluation, we used the Positive Predictive Value (Precision) and the Sensitivity (Recall) metrics. The former is the number of positive predictions divided by the total number of positive class values predicted, while the latter is the number of positive predictions divided by the total number of actual positive class values. To improve the experiments' accuracy, we applied extra filters. We noticed unequal values between the classes, class A being the minority class. We applied the SMOTE unsupervised filter to the minority [45]. In that way, the number of our instances increased from 2787 to 3150. Finally, we wanted to compare the performance between the 82 and the 96 feature dimensions, with and without using the SMOTE filter (Table 3).

Table 3. Precision and Recall of English-Greek corpus.

	82 Features		96 Features	
Class	**Precision**	**Recall**	**Precision**	**Recall**
Classifier: RandomForest-2687 instances				
A	49%	22%	53%	20%
B	46%	36%	47%	38%
C	50%	70%	52%	72%
Classifier: RandomForest_SMOTE-3150 instances				
A	77%	63%	77%	69%
B	44%	32%	48%	35%
C	50%	68%	53%	69%

It is noteworthy that when we apply Random Forest with the new additional features, there is an increase between 1% and 4%, i.e., we have better accuracy results for all classes, which is quite promising. Additionally, after applying the SMOTE filter, the minority class A has a much better prediction accuracy than before. We do not observe significant differences in the other two classes.

Furthermore, we wanted to see if this feature set would give good results in the English-Italian corpus. Therefore, we applied the same methodology to the English-Italian corpus, extracting 82 features and comparing the results with the 96 features (Table 4). We observed that the minority class also in this corpus is class A, so we applied again the SMOTE unsupervised filter to the minority class A [46] and our instances increased from 2745 to 3340.

Table 4. Precision and Recall of English-Italian corpus.

	82 Features		96 Features	
Class	**Precision**	**Recall**	**Precision**	**Recall**
Classifier: RandomForest-2745 instances				
A	25%	6%	29%	7%
B	36%	21%	34%	20%
C	46%	78%	47%	78%
Classifier: RandomForest_SMOTE-3340 instances				
A	71%	64%	71%	63%
B	36%	17%	38%	17%
C	48%	71%	48%	72%

The results using the 96 feature set and the SMOTE filter are generally better. In particular, we observed that the extra features improve the accuracy of class A (precision increase between 25% and 29%, recall increase between 6% and 7%) and we have similar results for the other two classes. The SMOTE filter also helps the minority class A (precision increase from 29% to 71%, and recall increase from 7% to 63%).

We decided to use the Deeplearning4j (DL4J) framework [43] using the Weka tool. DL4J is designed to handle large corpora. We used the previous 96 features and the SMOTE filter.

Firstly, we used k fold CV, which is a reliable method for testing our models, and a value of k = 10 is very common in the field of machine learning [47]. Then, we hold out a certain percentage (70%) of the data for training and the remaining part is used for testing (Table 5).

Table 5. Precision and Recall of English-Greek corpus using Deeplearning4j with 10 fold CV–Percentage split.

Classifier: Deeplearning4j				
	10 fold CV-3150 instances		**70% Per. split–945 instances**	
Class	**Precision**	**Recall**	**Precision**	**Recall**
A	67%	71%	65%	72%
B	45%	40%	50%	48%
C	52%	56%	52%	50%

As we observe, using Deeplearning4j with 10 fold CV we have satisfactory results, compared to Random Forest (Table 3), especially for class B. When we use the 70% Percentage split we notice a small performance improvement for class A, but we also have good results for class B (Precision increase from 45% to 50% and Recall increase from 40% to 48%), in contrast to Radom Forest results.

As can be seen in Table 6, we have tried the same architecture for the English–Italian corpus.

Table 6. Precision and Recall of English–Italian corpus using Deeplearning4j with 10 fold CV–Percentage split.

Classifier: Deeplearning4j				
	10 fold CV-3340 instances		**70% Per. split–1002 instances**	
Class	**Precision**	**Recall**	**Precision**	**Recall**
A	61%	64%	56%	51%
B	35%	31%	32%	37%
C	49%	51%	44%	42%

After applying Deeplearning4j to the English-Italian corpus with 10 fold CV we have satisfactory results, especially for class B. When we use the 70% Percentage split, we notice a small difference for class B.

There are many techniques for improving the classifier's performance. Many studies claim the importance of treating feature selection, as part of the learning process, in order to ensure a valid evaluation process [17]. When features are selected before applying the learning algorithm we have better results. Information Gain (IG) [19,48] and Correlation Feature Selection (CFS) [48] are language-independent feature selection methods that produce better accuracy. There are also language-dependent methods such as morphological normalization and words segmentation [49]. We have chosen the AttributeSelectedClassifier filter in Weka. It was applied in combination with the SMOTE supervised filter. We applied the Random Forest classifier with 65 iterations, 20 random features for constructing the trees, and we employed 70% percentage split as a test mode (Table 7).

Table 7. Precision and Recall-AttributeSelectedClasiffier.

AttributeSelectedClasiffier				
	English-Greek		**English-Italian**	
Class	**Precision**	**Recall**	**Precision**	**Recall**
A	73%	48%	76%	45%
B	48%	38%	36%	18%
C	49%	70%	44%	77%

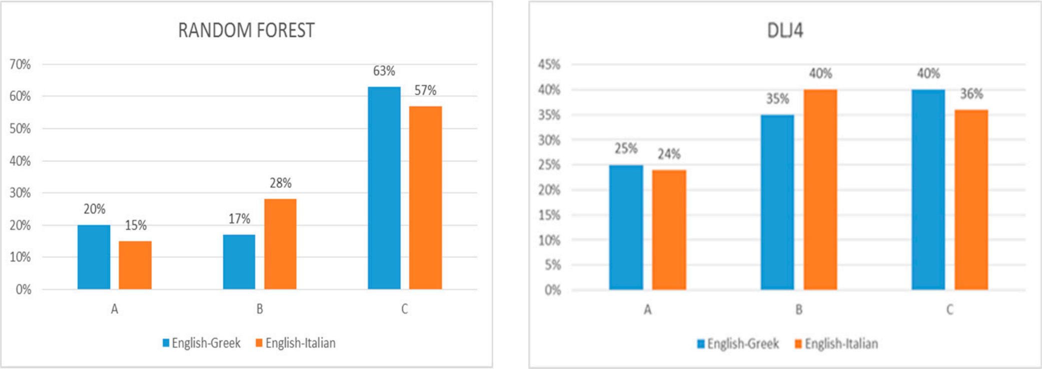
We generally observe better results, in contrast to Random Forest (Tables 3 and 4) and Dl4j (Tables 5 and 6), especially for class C in the EN-IT corpus. The features that are more effective for our model are features containing ratios. Also features that identify the presence of noise in a segment (for example the occurrence of 3 or more repeated characters) seem to be useful for prediction. The new features added in this paper seem to enclose valuable information for the model.

When the classifier is confused, it usually misclassifies instances from one neural model to the other (Table 8). Thus, (67% B→C, 14% C→B with Random Forest classifier in Italian, 56% B→C, 23% C→B with Random Forest in Greek) similar using the other architecture (DL4J) (41% B→C, 39% C→B for Italian and 41% B→C, 32% C→B for Greek). A much lower percentage to the statistical model (14% C→A, 15% B→A for Italian with Random Forest and 8% C→A, 9% B→A for Greek with Random Forest) analogous with the DL4J architecture (19% C→A, 22% B→A for Italian and 18% C→A, 11% B→A for Greek). This was also observed in our previous study [10]. This phenomenon is language- and model-independent.

Table 8. Confusion Matrix.

Class	**A**	**B**	**C**
	RF_GR-EN/IT-EN		
A	640/752	87/83	199/355
B	96/139	349/156	547/612
C	99/170	286/175	4846/898
	DL4J_GR-EN/IT-EN		
A	194/175	33/78	44/92
B	36/64	145/102	124/113
C	68/74	117/144	184/160

In total, we can see in Figure 1 that the majority of misclassified segments from classes A and B, were classified by Random Forest into class C (57% EN-GR and 63% EN-IT), for classes A and C (28% EN-GR and 17% EN-IT) were incorrectly classified into B. We observe low percentages (15% EN-GR and 20% EN-IT) from classes B and C to class A. The majority of misclassified instances from classes A and B were classified by DL4J into class C (40% EN-GR and 36% EN-IT), for classes A and C (35% EN-GR and 40% EN-IT) were incorrectly classified into B. We observe low percentages (25% EN-GR and 24% EN-IT) from classes B and C to class A.



(**a**) (**b**)

Figure 1. Total percentages for incorrectly classified segments for classifiers: (**a**) Random Forest; (**b**) DL4J.

We detected high percentages of misclassified segments from class B to class C. We reviewed some of these segment examples and found out that the classifier prefers, among the 3 segments (one for every class), the segment that having the most of the same words with the segment to be classified, or it prefers the segment that has preserved (and not translated) the abbreviations.

Another important thing is the execution time of each classifier through the process. The DL4J requires more time in contrast to RF. In our experiment, RF took approximately 5 mins for training and testing, whereas DL4J took 15 mins for this procedure, using gpu NVIDIA GTX 1080.

Comparison to Related Work

As it is mentioned earlier, there is limited work on using sequence to sequence comparisons for identifying useful data selection feedback for machine translation. To compare our experimental results with the state of the art [36], we ran additional experiments using SVM with different configurations (including three kernel functions) (Table 9).

Table 9. Precision and Recall–comparison SVM-RF.

English-Greek								
Class	**Precision**	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**	**Precision**	**Recall**
	SVMLinear		**SVMPoly**		**SVMRBF**		**RF**	
A	52%	38%	56%	49%	53%	37%	77%	**69%**
B	45%	33%	47%	33%	46%	32%	**48%**	**35%**
C	46%	65%	48%	64%	16%	67%	**53%**	**69%**
English-Italian								
A	45%	54%	46%	56%	45%	53%	**71%**	**63%**
B	25%	7%	37%	3%	36%	15%	**38%**	**17%**
C	41%	62%	42%	61%	41%	64%	**48%**	**72%**

Quite satisfactory results in terms of precision are observed using the SVM algorithm in our dataset. For this kind of data, we observed better accuracy results using RF than SVM. Probably a reason for this is that we had three translation outputs instead of one, and more segments (3150 English–Greek and 3340 English-Italian). In addition, we found that the additional proposed features improved the classification accuracy.

5. Discussion

This study approaches data selection as a classification problem and explores the idea by adding new features (distance-based) and an extra corpus, compared to our earlier work (ref).

Experimental results indicate an increase of up to 4%, for the three classes, using Random Forest with the new additional features. Furthermore, after applying the SMOTE filter, the minority class A reaches a much better prediction accuracy than before (Precision increases from 53% to 77% and Recall from 20% to 69%). We observe the same increase on the English–Italian corpus, class A (precision increase between 25% and 29%, recall increase between 6% and 7%) and we have similar results for the other two classes. The SMOTE filter also helps the minority class A in this corpus (precision increase from 29% to 71%, and recall increase from 7% to 63%). Finally, we have run additional experiments based on previous studies in this field, in order to compare our approach with others.

6. Conclusions

In this paper, we improved the classification accuracy of earlier experiments in automatic data selection for machine translation. Our experiments are based on two large parallel corpora databases, one in English–Greek and the other in English–Italian. Additionally, three translation models were used, one based on SMT, and the other two based on NMT. Furthermore, we extended the feature set: 96 features were taken into account. Three thousand, one hundred and fifty English-Greek and 3340 English-Italian parallel segments were annotated by four annotators (two Greek and two Italian language experts). Because of the genre of the data, we pre-process it before applying the classifiers. We also applied the SMOTE filter, in order to smooth the class imbalance in our datasets. The Radom Forest and a deep learning algorithm (DL4J) were used in our experiments. Because of the large number of features we employed attribute selection using the Weka toolset. Experimentation shows that the better prediction model for both datasets is the model Trans3, based on NMT machine translation, trained on in-domain data. For future work, we want to examine if extra additional language-dependent features, for example features based on grammatical categorization, could further help. Also, further investigation of the new approach to text classification–deep learning–could improve the results. Many possible combinations of neural networks, layer architectures and sizes, and other criteria can be used in order to improve the classification success rate. In this work, only a few combinations of layers were tested. Experiments using early stopping criteria could further improve classification accuracy. The use of other tools, like Tensorflow [50], may help in terms of accuracy and training time [51]. It should be noted that the main focus of the work described herein was not to research the optimal deep neural network architecture. The results with deep learning are initial, and by no means optimal. Changing our focus to building an optimal deep architecture is being investigated, other deep learning implementation tools (Tensorflow) [50] and architectures are being researched, and results (too premature to report yet) are more than promising.

Author Contributions: K.L.K., D.M. conceived of the idea, D.M., designed and performed the experiments, analyzed the results, drafted the initial manuscript and K.L.K., revised the final manuscript, supervision.

Funding: This research received no external funding.

Acknowledgments: The authors gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved both content and presentation. The authors would like to thank the two Italian language experts for the annotation, as well as TraMOOC project for the corpora used in the experiment.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Collobert, R.; Weston, J. A unified architecture for natural language processing: Deep neural networks with multitask learning. In Proceedings of the 25th International Conference on Machine learning, Helsinki, Finland, 5–9 July 2008; ACM: New York, NY, USA, 2008.
2. Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; Kuksa, P. Natural language processing (almost) from scratch. *JMLR* **2011**, *12*, 2493–2537.
3. Koehn, P.; Och, F.J.; Marcu, D. Statistical phrase-based translation. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, Edmonton, AB, Canada, 27 May–1 June 2003; ACL: Stroudsburg, USA, 2003.
4. Bentivogli, L.; Bisazza, A.; Cettolo, M.; Federico, M. Neural versus phrase-based machine translation quality: A case study. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; ACL: Stroudsburg, PA, USA, 2016.
5. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. In Proceedings of the 3th International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015; ICLR: San Diego, CA, USA, 2015.
6. Peris, Á.; Cebrián, L.; Casacuberta, F. Online Learning for Neural Machine Translation Post-editing. *arXiv*, 2017; arXiv:1706.03796.
7. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
8. Mnih, A.; Hinton, G.E. A scalable hierarchical distributed language model. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–11 December 2009; NIPS: San Diego, CA, USA, 2009.
9. Arora, R. Comparative analysis of classification algorithms on different datasets using WEKA. *IJCA* **2012**, *54*, 21–25. [CrossRef]
10. Mouratidis, D.; Kermanidis, K.L. Automatic Selection of Parallel Data for Machine Translation. In Proceedings of the IFIP International Conference on Artificial Intelligence Applications and Innovations, Rhodes, Greece, 25–27 May 2018; Springer: Berlin, Germany, 2018.
11. Kalchbrenner, N.; Blunsom, P. Recurrent continuous translation models. In Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing (EMNLP), Seattle, WA, USA, 18–21 October 2013; ACL: Stroudsburg, PA, USA, 2013.
12. Kyunghyun, C.; Bart, V.M.; Caglar, G.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; ACL: Stroudsburg, PA, USA, 2014.
13. Cho, K.; Van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. In Proceedings of the SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014; ACL: Stroudsburg, PA, USA, 2014.
14. Hill, F.; Cho, K.; Jean, S.; Devin, C.; Bengio, Y. Embedding word similarity with neural machine translation. *arXiv*, 2015; arXiv:1412.6448.
15. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; MIT Press: Cambridge, MA, USA, 2014.
16. Skansi, S. *Introduction to Deep Learning: From Logical Calculus to Artificial Intelligence*; Springer: Cham, Switzerland, 2018; pp. 135–145. ISBN 978-3-319-73003-5.
17. Smialowski, P.; Frishman, D.; Kramer, S. Pitfalls of supervised feature selection. *Bioinformatics* **2009**, *26*, 440–443. [CrossRef] [PubMed]
18. Bordes, A.; Chopra, S.; Weston, J. Question answering with subgraph embeddings. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; ACL: Stroudsburg, PA, USA, 2014.
19. Bhosale, D.; Ade, R. Feature Selection based Classification using Naive Bayes, J48 and Support Vector Machine. *IJCA* **2014**, *99*, 14–18. [CrossRef]

20. Qiang, G. An effective algorithm for improving the performance of Naive Bayes for text classification. In Proceedings of the Second International Conference on Computer Research and Development, Kuala Lumpur, Malaysia, 7–10 May 2010; IEEE: Los Alamitos, CA, USA, 2010.
21. Mohamed, W.N.H.W.; Salleh, M.N.M.; Omar, A.H. A comparative study of reduced error pruning method in decision tree algorithms. In Proceedings of the IEEE International Conference on Control System, Computing and Engineering (ICCSCE), Penang, Malaysia, 23–25 November 2012; IEEE: Piscataway, NJ, USA, 2012.
22. Phyu, T.Z.; Oo, N.N. Performance Comparison of Feature Selection Methods. *MATEC Web Conf.* **2016**, *42*, 1–4. [CrossRef]
23. Mulay, S.A.; Devale, P.R.; Garje, G.V. Decision tree based support vector machine for intrusion detection. In Proceedings of the International Conference on Networking and Information Technology (ICNIT), Manila, Philippines, 11–12 June 2010; IEEE: Piscataway, NJ, USA, 2010.
24. Bosch, A.; Zisserman, A.; Munoz, X. Image classification using random forests and ferns. In Proceedings of the IEEE 11th International Conference on Computer Vision (ICCV), Rio de Janeiro, Brazil, 14–20 October 2007; IEEE: Rio de Janeiro, Brazil, 2007.
25. Farabet, C.; Couprie, C.; Najman, L.; Lecun, Y. Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1915–1929. [CrossRef] [PubMed]
26. Koehn, P.; Hoang, H.; Birch, A.; Callison-Burch, C.; Federico, M.; Bertoldi, N.; Dyer, C. Moses: Open source toolkit for statistical machine translation. In Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, Prague, Czech Republic, 25–27 June 2007; ACL: Stroudsburg, PA, USA, 2007.
27. Pal, M. Random forest classifier for remote sensing classification. *IJRS* **2005**, *26*, 217–222. [CrossRef]
28. Xu, B.; Guo, X.; Ye, Y.; Cheng, J. An Improved Random Forest Classifier for Text Categorization. *J. Comput.* **2012**, *7*, 2913–2920. [CrossRef]
29. Chan, J.C.W.; Paelinckx, D. Evaluation of Random Forest and Adaboost tree-based ensemble classification and spectral band selection for ecotope mapping using airborne hyperspectral imagery. *Remote Sens. Environ.* **2008**, *112*, 2999–3011. [CrossRef]
30. Assunçao, F.; Lourenço, N.; Machado, P.; Ribeiro, B. DENSER: Deep Evolutionary Network Structured Representation. *arXiv*, 2018; arXiv:1801.01563. [CrossRef]
31. Snoek, J.; Rippel, O.; Swersky, K.; Kiros, R.; Satish, N.; Sundaram, N.; Patwary, M.; Prabhat, M.; Adams, R. Scalable bayesian optimization using deep neural networks. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 7–9 July 2015; JMLR: Lille, France, 2015.
32. Hinton, G.; Deng, L.; Yu, D. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Process. Mag.* **2012**, *29*, 82–97. [CrossRef]
33. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
34. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–8 December 2012; NIPS: San Diego, CA, USA, 2012.
35. Pighin, D.; Màrquez, L.; May, J. An Analysis (and an Annotated Corpus) of User Responses to Machine Translation Output. In Proceedings of the 8th International Conference on Language Resources and Evaluation, Istanbul, Turkey, 21–27 May 2012; European Language Resources Association (ELRA): Istanbul, Turkey, 2012.
36. Barrón-Cedeño, A.; Màrquez-Villodre, L.; Henríquez-Quintana, C.A.; Formiga-Fanals, L.; Romero-Merino, E.; May, J. Identifying useful human correction feedback from an on-line machine translation service. In Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, 3–9 August 2013; AAAI Press: Beijing, China, 2013.
37. Kordoni, V.; Birch, L.; Buliga, I.; Cholakov, K.; Egg, M.; Gaspari, F.; Georgakopoulou, Y.; Gialama, M.; Hendrickx, I.H.E.; Jermol, M.; et al. TraMOOC (Translation for Massive Open Online Courses): Providing Reliable MT for MOOCs. In Proceedings of the 19th annual conference of the European Association for Machine Translation (EAMT), Riga, Latvia, 30 May–1 June 2016; European Association for Machine Translation (EAMT): Riga, Latvia, 2016.
38. Sennrich, R.; Firat, O.; Cho, K.; Birch-Mayne, A.; Haddow, B.; Hitschler, J.; Junczys-Dowmunt, M.; Läubli, S.; Miceli Barone, A.; Mokry, J.; et al. Nematus: A toolkit for neural machine translation. In Proceedings of the EACL 2017 Software Demonstrations, Valencia, Spain, 3–7 April 2017; ACL: Stroudsburg, PA, USA, 2017.

39. Miceli-Barone, A.V.; Haddow, B.; Germann, U.; Sennrich, R. Regularization techniques for fine-tuning in neural machine translation. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 9–11 September 2017; ACL: Stroudsburg, PA, USA, 2017.
40. Rama, T.; Borin, L. Comparative evaluation of string similarity measures for automatic language classification. In *Sequences in Language and Text*; Mačutek, J., Mikros, G.K., Eds.; Walter de Gruyter: Berlin, Germany, 2015; Volume 69, pp. 203–231. ISBN 9783110394771.
41. Broder, A.Z. On the resemblance and containment of documents. In Proceedings of the Compression and Complexity of Sequences 1997, Washington, DC, USA, 11–13 June 1997; IEEE Computer Society: Washington, DC, USA, 1997.
42. Pouliquen, B.; Steinberger, R.; Ignat, C. Automatic identification of document translations in large multilingual document collections. In Proceedings of the International Conference Recent Advances in Natural Language Processing (RANLP), Borovets, Bulgaria, 10–13 September 2003; Recent Advances in Natural Language Processing (RANLP): Borovets, Bulgaria, 2003.
43. Deep Learning for Java. Available online: https://deeplearning4j.org/ (accessed on 8 October 2018).
44. Singhal, S.; Jena, M. A study on WEKA tool for data preprocessing, classification and clustering. *IJITEE* **2013**, *2*, 250–253.
45. Daskalaki, S.; Kopanas, I.; Avouris, N. Evaluation of classifiers for an uneven class distribution problem. *Appl. Artif. Intell.* **2006**, *20*, 381–417. [CrossRef]
46. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357. [CrossRef]
47. Kuhn, M.; Kjell, J. *Applied Predictive Modeling*; Springer: New York, NY, USA, 2013; p. 600.
48. Zhang, D.; Lee, W.S. Extracting key-substring-group features for text classification. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, 20–23 August 2006; ACM: New York, NY, USA, 2006.
49. Šilić, A.; Chauchat, J.H.; Bašić, B.D.; Morin, A. N-grams and morphological normalization in text classification: A comparison on a croatian-english parallel corpus. In Proceedings of the Portuguese Conference on Artificial Intelligence, Guimarães, Portugal, 3–7 December 2007; Springer: Berlin, Germany, 2007.
50. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. Tensorflow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, Savannah, GA, USA, 2–4 November 2016; OSDI: Savannah, GA, USA, 2016.
51. Kovalev, V.; Kalinovsky, A.; Kovalev, S. *Deep Learning with Theano, Torch, Caffe, Tensorflow, and Deeplearning4j: Which One Is the Best in Speed and Accuracy?* Springer: Berlin, Germany, 2016; p. 181. ISBN 978-3-319-54220-1.