applied sciences

MDPI

Article

Learning Eligibility in Cancer Clinical Trials Using Deep Neural Networks

Aurelia Bustos and Antonio Pertusa *

Pattern Recognition and Artificial Intelligence Group (GRFIA), Department of Software and Computing Systems, University Institute for Computing Research, University of Alicante, E-03690 Alicante, Spain; aurelia@medbravo.org

* Correspondence: pertusa@dlsi.ua.es

Received: 2 July 2018; Accepted: 19 July 2018; Published: 23 July 2018

Abstract: Interventional cancer clinical trials are generally too restrictive, and some patients are often excluded on the basis of comorbidity, past or concomitant treatments, or the fact that they are over a certain age. The efficacy and safety of new treatments for patients with these characteristics are, therefore, not defined. In this work, we built a model to automatically predict whether short clinical statements were considered inclusion or exclusion criteria. We used protocols from cancer clinical trials that were available in public registries from the last 18 years to train word-embeddings, and we constructed a dataset of 6M short free-texts labeled as eligible or not eligible. A text classifier was trained using deep neural networks, with pre-trained word-embeddings as inputs, to predict whether or not short free-text statements describing clinical information were considered eligible. We additionally analyzed the semantic reasoning of the word-embedding representations obtained and were able to identify equivalent treatments for a type of tumor analogous with the drugs used to treat other tumors. We show that representation learning using deep neural networks can be successfully leveraged to extract the medical knowledge from clinical trial protocols for potentially assisting practitioners when prescribing treatments.

Keywords: clinical trials; clinical decision support system; natural language processing; word embeddings; deep neural networks

1. Introduction

Clinical trials (CTs) provide the evidence needed to determine the safety and effectiveness of new medical treatments. These trials are the bases employed for clinical practice guidelines [1] and greatly assist clinicians in their daily practice when making decisions regarding treatment. However, the eligibility criteria used in oncology trials are too restrictive [2]. Patients are often excluded on the basis of comorbidity, past or concomitant treatments, or the fact they are over a certain age, and those patients that are selected do not, therefore, mimic clinical practice. This signifies that the results obtained in CTs cannot be extrapolated to patients if their clinical profiles were excluded from the clinical trial protocols. Given the clinical characteristics of particular patients, their type of cancer, and the intended treatment, discovering whether or not they are represented in the corpus of CTs that is available requires the manual review of numerous eligibility criteria, which is impracticable for clinicians on a daily basis.

The process would, therefore, greatly benefit from an evidence-based clinical decision support system (CDSS). Briefly, a CDSS could scan free-text clinical statements from medical records and output the eligibility of the patient in both completed or ongoing clinical trials based on conditions, cancer molecular subtypes, medical history, and treatments. Such a CDSS would have the potential advantages of (1) assessing the representation of the patient's case in completed studies to more

confidently extrapolate study results to each patient when prescribing a treatment in clinical practice, and (2) screening a patient's eligibility for ongoing clinical trials.

In this work, we constructed a dataset using the clinical trial protocols published in the largest public registry available, and used it to train and validate a model that is able to predict whether short free-text statements (describing clinical information, like patients' medical history, concomitant medication, type and features of tumor, such as molecular profiles, cancer therapy, etc.) are considered as *Eligible* or *Not Eligible* criteria in these trials. This model is intended to inform clinicians whether the results obtained in the CTs—and, therefore, the recommendation in the standard guidelines—can be confidently applied to a particular patient. The ultimate goal of this work is to assess whether representation learning using deep neural networks could be successfully applied to extract the medical knowledge available on clinical trial protocols, thus paving the way toward more involved and complex projects.

In the present work, the text was first preprocessed in order to construct training and validation sets. After extracting bigrams and word-embeddings (which are commonly used techniques used to generate semantic representations), we explored different state-of-the-art classification methods (FastText, Convolutional Neural Networks (CNN), Support Vector Machines (SVM), and k-Nearest Neighbors (kNN)). Finally, after validating and comparing the final classifiers, the model was further tested against an independent testing set.

The main contributions of this work are as follows:

- We propose a method to learn the eligibility for cancer clinical trials collected in last 18 years.
- Several classifiers (FastText, CNN, SVM, and kNN) are evaluated using word-embeddings for eligibility classification.
- Using learned deep representations, CNN and kNN (in this case, with average word-embeddings) obtain a similar accuracy, outperforming the other methods evaluated.
- Representation learning extracts medical knowledge in cancer clinical trials, and word-embeddings are suitable to detect tumor type and treatment analogies.
- In addition, word-embeddings are also able to cluster semantically related medical concepts.

The remainder of this paper is organized as follows. Section 2 provides a brief review of the methods related to the proposed work. Section 3 describes the dataset constructed and the methodology used, including the details employed to train the embeddings and the text classifiers. The evaluation results are detailed in Section 4, along with an analysis of the word-embeddings that were learned. Finally, Section 5 addresses our conclusions and future work.

2. Related Work

Artificial intelligence methods include, among others, rule-based systems, traditional machine learning algorithms, and representation learning methods, such as deep learning architectures.

Rule-based approaches in Natural Language Processing (NLP) seek to encode biomedical knowledge in formal languages in such a way that a computer can automatically reason about text statements in these formal languages using logical inference rules. MetaMap [3] is a widely known rule-based processing tool in the broader domain of biomedical language. It is a named-entity recognition system which identifies concepts from the Unified Medical Language System Metathesaurus in text (and MetaMap Lite [4]), the clinical Text Analysis and Knowledge Extraction System (cTAKES [5]), and DNorm [6]. Many systems have been built upon those tools. For example, in [7], an NLP System for Extracting Cancer Phenotypes from Clinical Records was built to describe cancer cases on the basis of a mention-annotation pipeline based on an ontology and a cTAKES system, and a phenotype summarization pipeline based on the Apache Unstructured Information Management Architecture (UIMA [8]).

With regard to the specific domain of clinical trials, prior work has focused on the problem of formalizing eligibility criteria using rule-based approaches and obtaining a computational model that could be used for clinical trial matching and other semantic reasoning tasks. Several languages could

be applied in order to express eligibility criteria, such as Arden syntax, Gello, and ERGO, among others. Weng et al. [9] presented a rich overview of existing options. SemanticCT allows the formalization of eligibility criteria using Prolog rules [10]. Milian et al. [11] applied ontologies and regular expressions to express eligibility criteria as semantic queries. However, the problem of structuring eligibility criteria in clinical trials so as to obtain a generalizable model still remains unsolved.

Devising formal rules and representations with sufficient complexity to accurately describe biomedical knowledge is problematic. As an example, the problem with discrete representations in biomedical taxonomies and ontologies is that they miss nuances and new words (e.g., it is impossible for them to keep up to date with the new drugs in cancer research). In addition, they are subjective, require human labor to create and adapt them, and it is hard to compute word similarity accurately. In order to solve these issues, machine learning methods can be trained to acquire this knowledge by extracting patterns from raw data.

In traditional machine learning, the features employed to train algorithms, such as SVMs or kNN, are usually given, while in representation learning (deep learning methods such as CNN), these features are learned [12]. Nonetheless, many factors regarding variation influence the semantic interpretation of the biomedical language, thus making it very difficult to extract high-level abstract features from raw text. Deep learning solves this central problem by means of representation learning by introducing representations that are expressed in terms of other simpler representations.

Deep learning models are beginning to achieve greater accuracy and semantic capabilities [13] than the prior state of the art with regard to various biomedical tasks, such as automatic clinical text annotation and classification. For example, a recent work [14] presented an attentional convolutional network that predicts medical codes from clinical text. It aggregates information from throughout the document using a CNN, and then uses an attention mechanism to select the most relevant segments for each of the thousands of possible codes. With regard to clinical text classification tasks, [15] proposed an approach with which to automatically classify a clinical text at a sentence level using deep CNNs to represent complex features.

To the best of our knowledge, this work is the first reported study to explore the use of deep learning techniques in order to directly achieve a semantic interpretation of eligibility criteria in clinical trials. In contrast to classic NLP approaches, to build the model, we omitted the constraints and limitations of previous steps, such as tokenization, stemming, syntactic analysis, named entity recognition (NER), the tagging of concepts to ontologies, rule definition, or the manual selection of features.

3. Materials and Methods

The system architecture is shown in Figure 1. Clinical trials statements were first preprocessed as described in Section 3.1. Then, word-embeddings were trained,as shown in Section 3.2, and classification to obtain the eligibility prediction is detailed in Section 3.3.

3.1. Dataset Building

A total of 6,186,572 labeled clinical statements were extracted from 49,201 interventional CT protocols on cancer (the URL for downloading this dataset is freely available at https://clinicaltrials.gov/ct2/results?term=neoplasm&type=Intr&show_dow). Each CT downloaded is an XML file that follows a structure of fields defined by an XML schema of clinical trials [16]. The relevant data for this project are derived from the intervention, condition, and eligibility fields written in unstructured free-text language. The information in the eligibility criteria—both exclusion and inclusion criteria—are sets of phrases and/or sentences displayed in a free format, such as paragraphs, bulleted lists, enumeration lists, etc. None of these fields use common standards, nor do they enforce the use of standardized terms from medical dictionaries and ontologies. Moreover, the language had the problems of both polysemy and synonymy.

The original data were exploited by merging eligibility criteria together with the study condition and intervention, and subsequently transforming them into lists of short labeled clinical statements that consisted of two extracted features (see example in Figure 2), the label (Eligible or Not Eligible), and the processed text that included the original eligibility criterion merged with the study interventions and the study conditions. These processes are detailed in the following section.

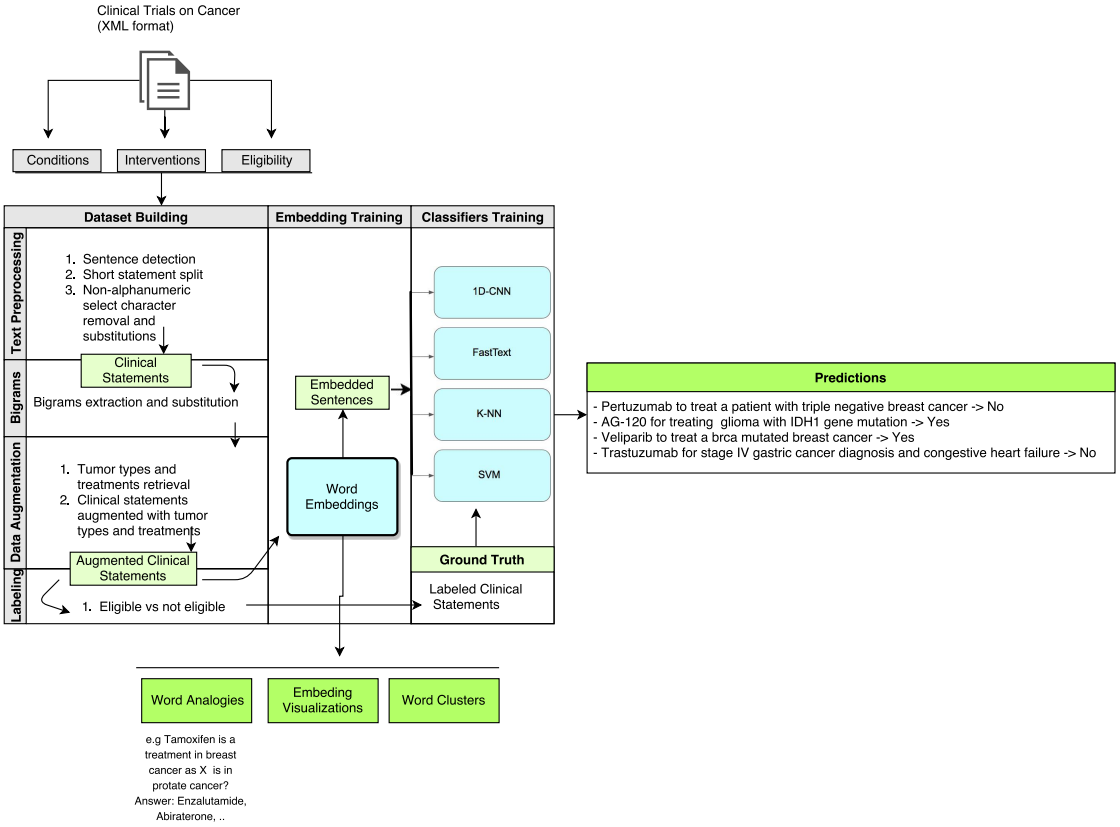


Figure 1. System Architecture: The objective of the final model is to predict whether or not short clinical statements concerning the type of tumor, including the molecular profile, the oncologic treatment, the medical history, or concomitant medication, were included in clinical trials.

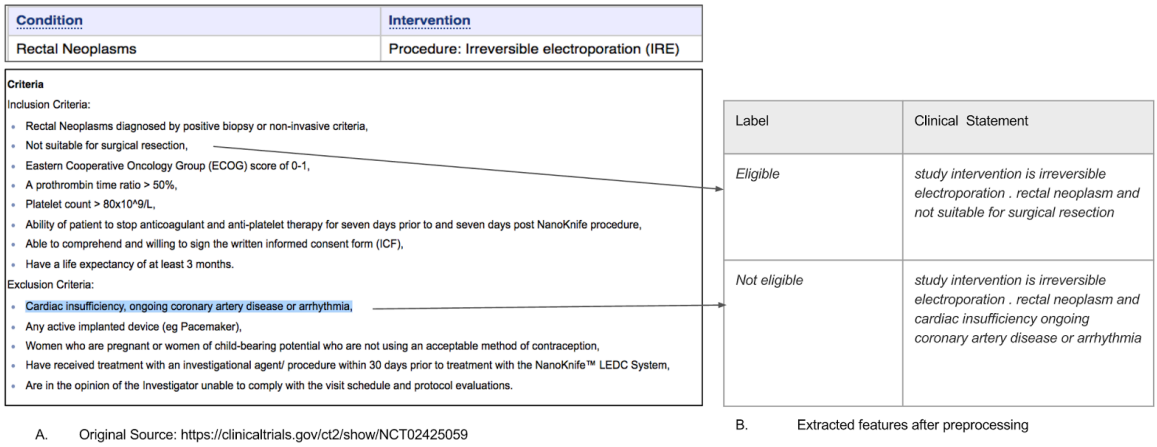


Figure 2. Extraction of labeled short clinical statements. The two example criteria indicated with the arrows were extracted from their original source, preprocessed, and labeled.

3.1.1. Text Preprocessing

We transformed all the eligibility criteria into sequences of plain words (and bigrams) separated by a whitespace. Each eligibility criterion was augmented with information concerning study intervention and cancer type, as illustrated in Figure 2. This was done by:

- Splitting text into statements: The implementation took into consideration different kinds of bullets and lists, and not mistakenly splitting into sentences common abbreviations used in mutations and other medical notations, which include dots, semicolons, or hyphens.
- Removing punctuation, whitespace characters, all non-alphanumeric symbols, separators, and single-character words from the extracted text. All the words were lowercase. We decided not to remove stop words because many of them, such as "or", "and", "on", were semantically relevant to the clinical statements.
- Transforming numbers, arithmetic signs (+/−), and comparators (>, <, =, ...) into text.

In order to filter out nonrelevant or useless samples, we discarded all the studies where the conditions did not include any of the tokens or suffixes in "cancer", "neoplasm", "oma", or "tumor". Given that the presence or absence of redundancy in eligibility criteria, both intra- or interstudy, is relevant information to be learned by the model, we did not filter out samples by this criteria, so that the original redundancy distribution was preserved in the dataset. Because preprocessing the entire dataset is a costly process, for those readers interested in reproducing this work but would like to skip the preprocessing steps, we made publicly available a random preprocessed subsample (https://www.kaggle.com/auriml/eligibilityforcancerclinicaltrials, at section Data, Download all) of 10^6 samples (for details, see Section 3.1.1).

3.1.2. Bigrams

In the scope of this work, we define bigrams as commonly found phrases that are very frequent in medicine. Some frequent bigrams were detected and replaced in the text. Bigrams can represent idiomatic phrases (frequently co-occurring tokens) that are not compositions of the individual words. Feeding them as a single entity to the word-embedding rather than each of its word separately, therefore, allows these phrase representations to be learnt. In our corpus, excluding common terms, such as stop words, was unnecessary when generating bigrams. Some examples of bigrams in this dataset are: sunitinib malate, glioblastoma multiforme, immuno histochemistry, von willebrand, dihydropyrimidine dehydrogenase, li fraumeni, etc.

Phrase (collocation) detection was carried out using the GenSim API [17]. The threshold parameter defines which phrases will be detected on the basis of their score. The score formula applied [18] is:

$$score(w_i, w_j) = \frac{count(w_i, w_j) - \delta}{count(w_i) \cdot count(w_j)} \quad (1)$$

For this dataset, after several tests, the most suitable threshold was set to 500, and the discounting coefficient δ was based on a min count of 20. The discounting factor prevents the occurrence of too many phrases consisting of very infrequent words. A total of 875 different bigrams were retrieved from the corpus and substituted in the text.

3.1.3. Data Augmentation

In this work, data augmentation consisted of adding the cancer types and interventions being studied to each criterion using statements such as: "patients diagnosed with [cancer type]".

In the case of CTs that studied multiple cancer types or interventions, we replicated each criterion for each intervention and condition, increasing the number of prototypes.

3.1.4. Labeling

After preprocessing and cleaning the data, the available set had 6,186,572 short clinical statements containing a total of 148,038,397 words. The vocabulary consisted of 49,222 different words. Each statement had in average 23.9 words with a range from 6 to 439 words. The distribution of number of words by statement had a mean = 23.9, variance = 171.3, skewness = 3.13, and kurtosis = 21.05.

For the ground truth, we automatically labeled the clinical statements—previously processed from the eligibility criteria, study conditions, and interventions—as "Eligible" (inclusion criterion) or "Not Eligible" (exclusion criterion) on the basis of:

- Their position in relation to the sentences "inclusion criteria" or "exclusion criteria", which usually preceded the respective lists. If those phrases were not found, then the statement was labeled "Eligible".
- Negation identification and transformation: negated inclusion criteria starting with "no" were transformed into positive statements and labeled "Not Eligible". All other possible means of negating statements were expected to be handled intrinsically by the classifier.

The classes were unbalanced, and only 39% of them were labeled as Not Eligible, while 61% were labeled as Eligible. As the dataset was sufficiently large, we used random balanced undersampling [19] to correct it, resulting in a reduced dataset with 4,071,474 labeled samples. The "eligibility" variable containing the text for each criterion, as expected in NLP, has a highly sparse distribution and only 450 entries were repeated.

3.2. Embedding Training

We used two different approaches (FastText [20] and Gensim [17]) to generate Word2Vec embeddings based on the skip-gram and CBOW models [21]. Word2vec [18] is a predictive model that uses raw text as input and learns a word by predicting its surrounding context (continuous BoW model) or predicts a word given its surrounding context (skip-gram model) using gradient descent with randomly initialized vectors. In this work, we used the Word2Vec skip-gram model. The main differentiating characteristic of FastText embeddings, which apply char *n*-grams, is that they take into account the internal structure or words while learning word representations [20]. This is especially useful for morphologically rich languages. FastText models with char *n*-grams perform significantly better when carrying out syntactic tasks than semantic tasks, because the syntactic questions are related to the morphology of the words.

We explored different visualizations projecting the trained word-embeddings into the vector space (Sections 4.5.1 and 4.5.2), grouped terms in semantic clusters (Section 4.5.3), and qualitatively evaluated the embeddings according to their capacity to extract word analogies (Section 4.5.4).

Table 1 shows the best hyperparameters found to generate 100 dimensional embeddings with the FastText and Gensim Word2Vec models. A random search strategy [22] was used in order to optimize the values of these parameters. The Gensim model was trained with three workers on a final vocabulary of 22,489 words using both skip-grams and CBOW models.

Table 1. Word2Vec hyperparameters using FastText and GenSim. Optimization was performed using random search [22].

Hyperparameter	FastText	GenSim
Learning rate	0.025	0.025
Size of word vectors	100	100
Size of the context window	5	5
Number of epochs	5	5
Min. number of word occurrences	5	5
Num. of negative sampled	5	5
Loss function	negative sampling	negative sampling
Sampling threshold	10^{-4}	10^{-3}
Number of buckets	2,000,000	
Minimum length of char *n*-gram	3	
Maximum length of char *n*-gram	6	
Rate of updates for the learning rate	100	

3.3. Classifier Training

Once the word-embeddings were extracted, the next stage consisted of sentence classification. For this, we explored four methods: Deep Convolutional Neural Networks [13] with or without pre-trained word-embeddings at the input layer, FastText [23], Support Vector Machines (SVM), and k-Nearest Neighbors (kNN).

Learning curves were built for all models with increasing dataset sizes (1K, 10K, 100K, 1M, and 4.07 M samples). Each dataset was sampled from the full dataset, applying random balanced sampling so that, for each resulting dataset, both label classes ("Eligible" and "Not Eligible") had the same number of samples. We split each dataset into 80% samples for the training set and 20% for the test set. A standard 5-fold cross-validation was then performed for each dataset size.

Because the accuracy concerning sentence classification depends on the dataset evaluated and we were unable to find any previous reports that used the present corpus for text classification, there are no clearly defined benchmarks with which to perform a comparison.

For example, in different domains, the reported accuracy for classifying the "Hacker News" posts into 20 different categories using a similar method was 95%, while in the case of "Movie reviews", the reported performance was 81.5% [24]. In the medical domain, a high-performance model is potentially useful in a CDSS. Using previously published computer-aid systems and related work [25–27] as a basis, we defined the minimum target as an accuracy of 90%, and a Cohen's Kappa with a minimum of [0.61–0.80] for substantial agreement, or [0.81–1] for an almost perfect agreement [28].

3.3.1. FastText

FastText [20,23] for supervised learning is a computationally efficient method that starts with an embedding layer which maps the vocabulary indexes into d dimensions or, alternatively, it can use pre-trained word vectors. It then adds a global average pooling layer, which averages the embeddings of all the words in the sentence. Finally, it projects it onto a single unit output layer and squashes it with a sigmoid.

3.3.2. Convolutional Neural Network

In the first experiment, the pre-trained word-embeddings were used as the input for the 1D CNN model, which has a final dense output layer. In a different experiment, we also trained the word-embeddings for our classification task from scratch. As the training data was sufficiently large and the vocabulary coverage was also appropriate for the cancer research domain, it was expected that the model would benefit from training the embeddings in this particular domain.

We used the Keras [29] library to build a CNN topology (see Table 2), inspired by the text classifier model for the 20 Newsgroup datasets [30]. After the necessary adaptations, we followed the steps shown below:

1. Convert all the sentences in the dataset into sequences of word indexes. A *word index* is simply an integer identifier for the word. We considered only the top 20,000 most commonly occurring words in the dataset, and truncated the sequences to a maximum length of 1000 words.
2. Shuffle, stratify, and split sequences of word indexes into training (80%) and validation sets (20%).
3. Prepare an *embedding matrix* which contains at index i the embedding vector for the word from index i. We loaded this embedding matrix into an embedding layer which was frozen (i.e., its weights, the embedding vectors, were not updated during training).
4. A 1D CNN ending in a Softmax layer with two classes was built on top of it.
5. During training, the data were shuffled with random seed before each epoch (we used 10 epochs).

Table 2. CNN topology used in this work. The architecture was chosen after evaluating the accuracy on the test set using different kernel sizes, number of layers, activation functions, etc.

Layer	Description
Input	1000 × 100 dimensional embedded word sequences
Convolution	128 5 × 1 convolutions with stride 1 and ReLu activation
Max Pooling	5 × 1 max pooling with stride 1
Convolution	128 5 × 1 convolutions with stride 1 and ReLu activation
Max Pooling	5 × 1 max pooling with stride 1
Convolution	128 5 × 1 convolutions with stride 1 and ReLu activation
Max Pooling	35 × 1 max pooling with stride 1
Fully Connected	128 fully connected layer with ReLu activation
Fully Connected	2 fully connected layer with Softmax activation

3.3.3. SVM

A support vector machine [31] constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training data points of any class (so-called functional margin), since, in general, the larger the margin, the lower the generalization error of the classifier. We trained an SVM model with the following hyper-parameters selected using exhaustive grid-search optimization: penalty parameter C or the error term = 1, kernel = rbf, kernel gamma coefficient = 1, shrinking heuristic = True, tolerance for stopping criterion = 0.001.

For each short clinical statement, its pre-trained word-embeddings (obtained with FastText using the skip-gram model, as explained in Section 3.2) were used to calculate an average vector of dimension 100 for each clinical statement. Therefore, given a statement, an average vector of word-embeddings serves as input to the SVM. This representation was chosen to reduce the dimensionality of the input data.

3.3.4. kNN

Neighbors-based classification is a type of instance-based learning or nongeneralizing learning: it does not attempt to construct a general internal model, but simply stores instances of the training data. Classification is computed from a simple majority vote of the nearest neighbors of each point: a query point is assigned the data class which has the most representatives within the nearest neighbors of the point. The same input data used for SVM were evaluated using kNN. We trained a kNN model with the following hyper-parameters selected using exhaustive grid-search optimization: number of neighbors = 3, uniform weight for all points in each neighborhood, and Euclidean distance metric.

4. Results

4.1. Metrics

The performance of the models was calculated using the F-measure (F_1), precision and recall, the confusion matrix, and the coefficient of agreement.

Precision (also called positive predictive value) is the fraction of retrieved instances that are relevant. Recall (also known as sensitivity) is the fraction of relevant instances that are retrieved. Precision, Recall, and F_1, which is the harmonic mean of precision and sensitivity, are calculated as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$
$$F_1 = 2 \times \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

where TP (true positives) denotes the number of correct predictions, FP (false positives) is the number of "Not Eligible" labels wrongly predicted as "Eligible", and FN (false negatives) is the number of "Eligible" labels wrongly declared as "Not Eligible".

Cohen's Kappa (κ) is a statistic that measures the inter-rater agreement for qualitative (categorical) items. It is generally considered to be a more robust measure than a simple percent agreement calculation, since κ takes into account the possibility of the agreement occurring by chance [32]. It is calculated as:

$$\kappa = \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e} \tag{2}$$

where p_o is the relative observed agreement among raters, and p_e is the hypothetical probability of a chance agreement, using the observed data to calculate the probabilities of each observer randomly yielding each category.

4.2. Model Evaluation and Validation

All the models were evaluated using different configurations of hyper-parameters, and the best results obtained for each classifier are given in Table 3. This section details the classifier settings to get these results, and analyzes the learning curves that can be seen in Figure 3.

Table 3. Overall results on the validation set for all the classifiers using a dataset of 10^6 samples and the full dataset (4.1×10^6) samples. Both experiments were performed using 20% of the prototypes for validation and 80% for training. The best results are marked in bold.

Classifier	Dataset Size	Precision	Recall	F_1	Cohen's κ
FastText	10^6	0.88	0.86	0.87	0.75
	4.1×10^6	0.89	0.87	0.88	0.76
CNN	10^6	0.88	0.88	0.88	0.76
	4.1×10^6	0.91	0.91	0.91	0.83
SVM	10^6	0.79	0.79	0.79	0.57
	4.1×10^6	0.79	0.79	0.79	0.58
kNN	10^6	0.92	0.92	0.92	0.83
	4.1×10^6	0.93	0.93	**0.93**	**0.84**

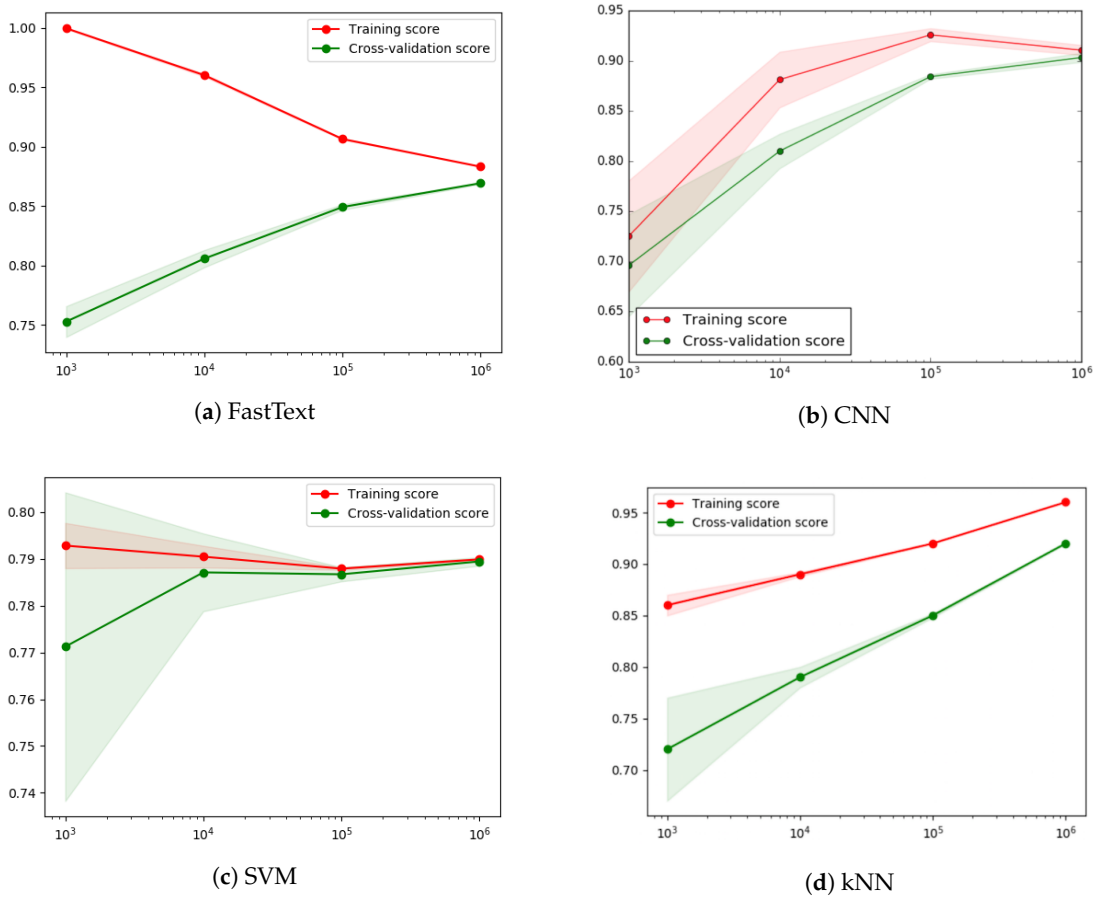


Figure 3. Learning curves with 5-fold cross-validation on the FastText, Convolutional Neural Network (CNN), Support Vector Machine (SVM), and k-Nearest Neighbors (kNN) classifiers. Horizontal axes show the total training samples, whereas vertical axes are the F_1-score. The green region represents the standard deviation of the models.

4.2.1. FastText Classifier Results

In order to choose the parameters for the FastText model, we compared the F_1 between successive experiments. Using a random search [22] strategy for hyperparameter search, the best results were obtained with 100 dimensions and a learning rate of 0.1, as shown in Table 4.

We also tested the predictive performance of this model when using or not using pregenerated bigrams, but there was no significant impact on the results, as shown in Figure 4.

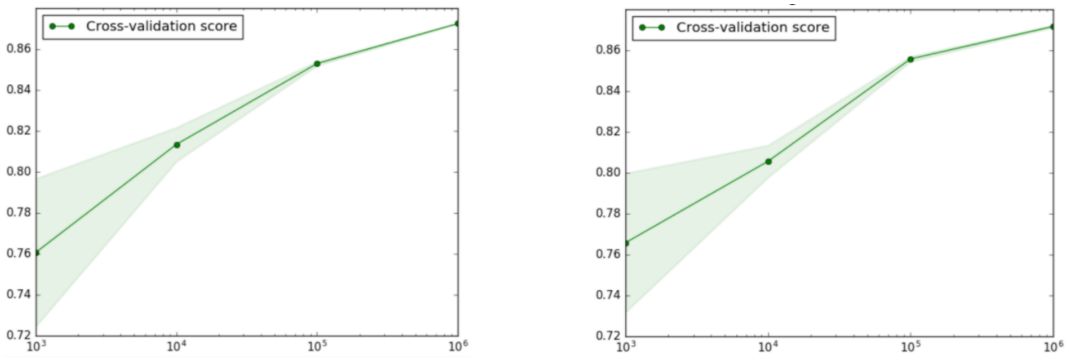
The F_1 achieved when using 10^6 samples (800K for training) was 0.87 (see Table 3). The Cohen's Kappa coefficient of agreement between the predicted and the true labels in the validation set was $\kappa = 0.75$, which is regarded as a substantial agreement. The results did not improve significantly when using the full dataset of 4.1×10^6 samples. In this case, the F_1 achieved was 0.88 with a Cohen's Kappa coefficient $\kappa = 0.76$.

The learning curve (Figure 4) shows the evolution of the F_1 during training when the number of training samples was increased from 800 to 800K. The curve converged with the score obtained in the training sample to a maximum of 0.88 when using the full dataset, as shown in Table 3. The validation score converges with the training score, and the estimator does not benefit much from more training data, denoting a bias error. It has been reported that the phenomenon of not being able to increase the performance with additional data can be overcome with the use of deep learning models applied to complex problems, in contrast to a fast but thin architecture such as FastText (as will be proved later when using CNNs). On the contrary, the model did not suffer from a variance error. Cross-validation

was used to assess how well the results of the model generalized to unseen datasets and obtained robust average validation results with a decreasing standard deviation over the k folds (Figure 3a).

Table 4. FastText classifier hyper-parameters. Optimization was performed using random search [22].

Hyper-Parameter	Value
Learning rate	0.1
Size of word vectors	100
Size of the context window	5
Number of epochs	100
Minimum number of word occurences	1
Number of negatives sampled	5
Loss function	Softmax
Minimum length of char n-gram	0
Maximum length of char n-gram	0
Maximum length of word n-gram	1
Sampling threshold	10^{-4}
Rate of updates for the learning rate	100
Use of pre-trained word vectors for supervised learning	Yes



(**a**) With bigrams (**b**) Without bigrams

Figure 4. Learning curves with 5-fold cross-validation on the FastText classifier using as input pre-trained word-embeddings learned (**a**) with bigrams and (**b**) without bigrams. The green region represents the standard deviation of the model.

4.2.2. CNN Classifier Results

The hyper-parameters used to train the CNN model are shown in Table 5. The results obtained when using both Gensim and FastText generated embeddings were studied, and the F_1 obtained was similar when using or not using pre-trained word-embeddings. Only the number of dimensions and epochs had a great impact on the performance and the computational cost of the model.

With regard to the batch size, sizes of 1, 10, 64, 128, and 512 were investigated and, as expected, the higher the value, the greater the computational efficiency. The noisiness of the gradient estimate was reduced in batch sizes by using higher values. This can be explained by the fact that updating by one single sample is noisy when the sample is not a good representation of all the data. We should consider a batch with a size that is representative of the whole dataset. For values higher than 128, the predictive performance deteriorated in earlier epochs during training and, therefore, we chose a value of 128. In fact, it has been reported that the loss function landscape of deep neural networks is such that large-batch methods are almost invariably attracted to regions with sharp minima [33] and that, unlike small-batch methods, they are unable to escape the basins of these minimizers. When using a larger batch, there is consequently a significant degradation in the quality of the model, as measured by its ability to generalize.

Table 5. CNN classifier hyper-parameters.

Hyper-Parameter	Value
Batch size	128
Learning rate	0.001
Size of word vectors	100
Number of epochs	10
Max number of words	20,000
Max sequence length	1000
Loss function	Categorical cross-entropy
Optimizer	RMSProp
RMSProp rho	0.9
epsilon	10^{-8}
decay	0

The CNN learning curve (Figure 3b) shows that the network is capable of generalizing well and that the model is robust. Unlike that which occurs with the FastText classifier, no overfitting is produced when the dataset is small.

Nonetheless, it also had a bias error, but, in this case, the model achieved higher scores for both the training and the validation sets, converging to a maximum $F_1 = 0.91$, beyond which adding more data does not appear to be beneficial.

One additional difference with the FastText learning curve is that the CNN model needs more data to learn, in comparison with FastText. This is reflected by the fact that the CNN model was underfitting and not properly learning for a sample size 10^3 with a validation score of only 0.72, while for FastText and a sample size 10^3, the model was clearly overfitting with a training score close to 1.

The model of the whole dataset, using 3,257,179 training examples, bigrams, and pre-trained word-embeddings, eventually yielded an accuracy of 0.91 for the validation set comprising 814,295 samples. The coefficient of agreement between the predicted and the true labels in the validation set was $\kappa = 0.83$ (see Table 3), which is regarded as an almost perfect agreement and implies that the model is reliable.

4.3. SVM Classifier Results

The learning curve (Figure 3c) shows the evolution of the F_1 during training when the number of training samples was increased from 800 to 800K. The curve converged with the score obtained in the training sample to a maximum of 0.79 when using the full dataset, as shown in Table 3. The validation score converges with the training score, and the estimator does not benefit much from more training data, denoting a bias error.

The F_1 achieved when using 10^6 samples (800K for training) was 0.79 (see Table 3). The Cohen's Kappa coefficient of agreement between the predicted and the true labels in the validation set was $\kappa = 0.57$. The results did not improve when using the full dataset of 4.1×10^6 samples. In this case, the F_1 achieved was 0.79 with a Cohen's Kappa coefficient $\kappa = 0.58$.

4.4. kNN Classifier Results

The learning curve (Figure 3d) shows the evolution of the F_1 during training when the number of training samples was increased from 800 to 800K. The validation curve with a maximum of 0.92 still did not reach the training score obtained in the 800K training sample and further reached 0.93 in the full dataset, as shown in Table 3. This estimator benefited the most, compared with the other models, from more training data. Nonetheless, as expected, the computational cost on prediction time was expensive, and using 10^6 samples was equivalent to 16 core-hours of CPU.

The F_1 achieved when using 10^6 samples (800K for training) was 0.92 (see Table 3). The Cohen's Kappa coefficient of agreement between the predicted and the true labels in the validation set was $\kappa = 0.83$, which is regarded as an almost perfect agreement. The results did not improve significantly

when using the full dataset of 4.1×10^6 samples. In this case, the F_1 achieved was 0.93 with a Cohen's Kappa coefficient $\kappa = 0.84$.

4.4.1. Evaluation Using a Clinical Practice Simulation

Finally, in order to assess the potential of the proposed approach as a clinical decision support system, we checked its performance using a clinical practice simulation. The two final models were, therefore, further tested with unseen inputs consisting of a small set (50 samples) of short clinical statements that would be used in routine clinical practice. Although the test size is too small to be able to draw meaningful conclusions, the models yielded very promising results with an accuracy of 0.88 and $\kappa = 0.76$. This favors the hypothesis that it would be possible to generalize such a model to a different source of data (i.e., routine clinical practice notes) beyond clinical trial protocol eligibility criteria texts, which was the source used to build and validate it.

Some examples of correctly classified statements that would require an expert knowledge of oncology to judge whether or not they are cases being studied in available clinical trials (Yes/No) are shown below.

Lapatinib to treat breast cancer with brain metastasis → Yes;
Pertuzumab to treat breast cancer with brain metastasis → No;
CAR to treat lymphoma → Yes;
TCR to treat breast cancer → No.

The performance achieved with the CNN classifier fits expectations with an $F_1 = 0.91$ and an almost a perfect agreement, outperforming the FastText results. We can, therefore, conclude that it is possible to address the problem of predicting whether or not short clinical statements extracted from eligibility criteria are considered eligible in the available corpus of cancer clinical trials.

4.5. Word-Embeddings

The word-embeddings are an interesting part of this work. Adding pre-trained embeddings to the classifiers did not alter the classification results. However, the embeddings were, in themselves, sufficiently interesting to be qualitatively assessed and discussed using word space visualizations.

4.5.1. t-SNE (t-Distributed Stochastic Neighbor Embedding) Representation of a Subset of Words

The word-embeddings obtained with FastText, in which each word is represented in a 100-dimensional space, can be used as a basis on which to visualize a subset of these words in a reduced space. We use t-Distributed Stochastic Neighbor Embedding (t-SNE [34]) for this purpose, which is a dimensionality reduction method that is particularly well suited to the visualization of high-dimensional datasets. The objective of this algorithm is to compute the probability distribution of pairs of high-dimensional samples in such a way that similar prototypes will have a high probability of being clustered together. The algorithm subsequently projects these probabilities into the low-dimensional space and optimizes the distance with respect to the sample's location in that space.

We defined those words from the complete corpus that we wished to analyze (as it is not possible to visualize all 26,893 words), and obtained the vectors of these words. The t-SNE representation in Figure 5 shows two aspects: on the one hand, the words are grouped by semantic similarities, and on the other, the clusters seem to follow a spatial distribution in different regions in a diagonal direction from intrinsic/internal to extrinsic/external concepts with respect to the human body: [G5] body organs → [G4] body symptoms → [G3] infections, cancer and other diseases → [G1,G2] treatments.

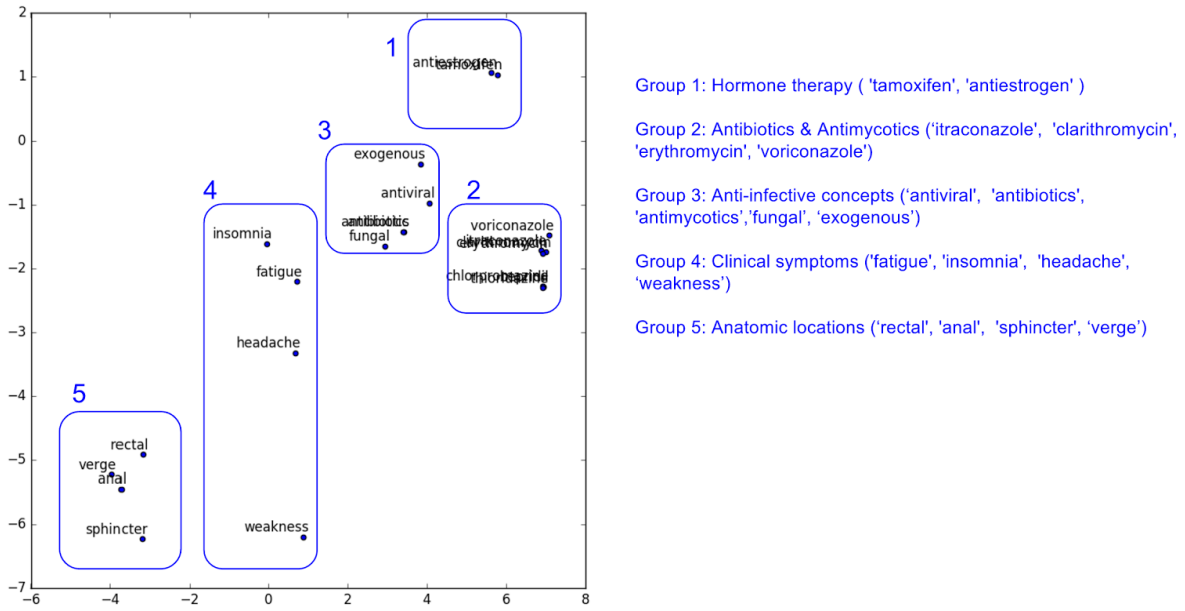


Figure 5. Word-embeddings projected into a reduced space with t-Distributed Stochastic Neighbor Embedding (t-SNE).

4.5.2. Interactive Visualization of the Whole Set of Words

TensorBoard from TensorFlow [35] provides a built-in visualizer, called the Embedding Projector, for the interactive visualization and analysis of high-dimensional data. The Word2Vec embeddings obtained with Gensim were converted into Tensorflow 2D tensor and metadata formats for embedding visualization.

Figure 6 shows an example of these results when using the word "ultrasound" as a query. We can appreciate that the 87 nearest points to ultrasound were all related to explorations, and mainly medical imaging. The nearest neighbor distances are also consistent when using other concepts. For example, Table 6 shows that the model successfully extracted hormonal therapies from breast cancer as the t-SNE nearest neighbors to "Tamoxifen".

Table 6. Nearest neighbors of "Tamoxifen" using Euclidean distance on the embedding t-SNE space. All of them are hormonal therapies.

Word	Distance
Raloxifene	0.569
Letrozole	0.635
Anastrozole	0.656
Fulvestrant	0.682
Arimidex	0.697
Antiandrogens	0.699
Exemestane	0.715
Aromatase	0.751
Antiestrogens	0.752
Toremifene	0.758
Serm	0.760
Estrogens	0.769
Agonists	0.773

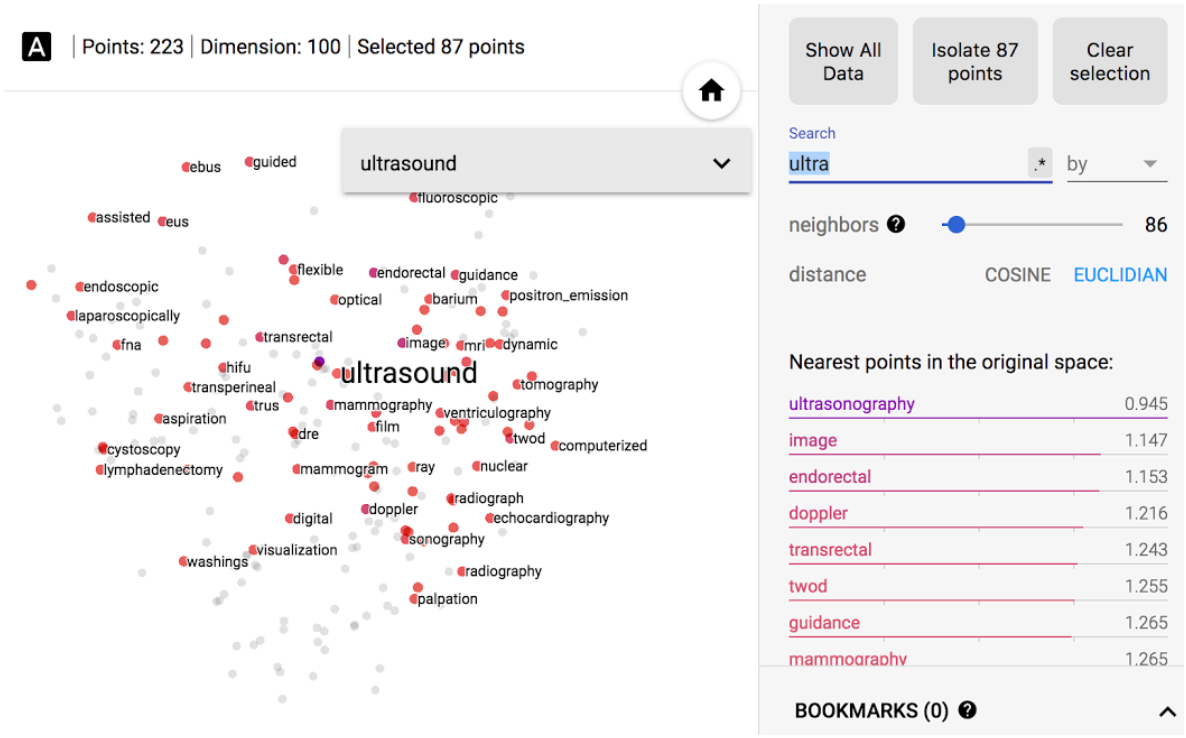


Figure 6. Search for "Ultrasound" on the Tensorboard Embedding Projector.

4.5.3. Word Clusters

We also used the resulting word vectors to generate word clusters fitting a *k*-means model [36]. The number of clusters were estimated by applying a reduction factor of 0.1 to the total number of words to be read (maximum 10,000). The implementation and resulting clusters can be found at https://github.com/auriml/capstone. Upon sampling 20 clusters at random, a total of 16 were judged to be relevant as to whether their words were syntactically or semantically related. Some examples are shown in Table 7.

Table 7. Samples of clustered words.

mri, scan, imaging, radiographic, magnetic, resonance, scans, abdomen, radiological, mr, radiologic, image, technique, images, perfusion, sectional, weighted, spectroscopy, mris, dce, imaged, lp, neuroimaging, volumetric, mrs, multiparametric, mrsi, imagery	pelvis, skull, bones, skeleton, femur, ribs, sacrum, sternum, sacral, lfour, rib, humerus	pulmonary, respiratory, obstructive, asthma, copd, restrictive, emphysema, bronchiectasis, bronchodilator, bronchitis, bronchospasm, pneumothorax, ssc, bronchopulmonary, cor, expired, onel, congestion, airflow	abuse, alcohol, substance, dependence, alcoholism, addiction, dependency, illicit, recreational, user, illegal, misuse, abusers

Note that medical abbreviations, such as *lfour* (L4), *mri* (Magnetic Resonance Imaging), or *copd* (Chronic Obstructive Pulmonary Disease) were correctly clustered.

4.5.4. Word Analogies

The word vectors generated were also useful for accurately resolving analogy problems, such as "Tamoxifen is used to treat breast cancer as *X* is used to treat prostate cancer?". To find the top-N most similar words, we used the multiplicative combination "3CosMul" objective proposed by Levy [37]:

['tamoxifen' − 'breast + 'prostate'] ≈ [('enzalutamide', 0.998), ('antiandrogens', 0.972), ('abiraterone', 0.952), ('finasteride', 0.950), ('zoladex', 0.946), ('adt', 0.933), ('dutasteride', 0.927), ('acetate', 0.923), ('flutamide', 0.916), ('leuprolide', 0.910)]

These are, in fact, very precise results, because all these terms belong to the hormone-therapy family of drugs which are specifically used to treat prostatic cancer, and are the equivalents of tamoxifen (hormone-therapy) for breast cancer. In other words, the model learned the abstract concept "hormone-therapy" as a family of drugs and was able to apply it distinctively depending on the tumor type.

5. Conclusions

In this work, we have trained, validated, and compared various classifiers (FastText and a CNN with pre-trained word-embeddings, kNN, and SVM) on a corpus of cancer clinical trial protocols (www.clinicaltrials.gov). The models classify short free-text sentences describing clinical information (medical history, concomitant medication, type and features of tumor, such as molecular profile, cancer therapy, etc.) as eligible or not eligible criteria for volunteering in these trials. SVM yielded the lowest accuracy results, and kNN obtained top accuracy performance similar to the CNN model, but it had the lowest computational performance. Particularly, the high accuracy achieved with kNN is the immediate consequence of using as input a highly efficient clinical statement representation which is based on averaged pre-trained word-embeddings. A possible reason for this is that the kNN accuracy relies almost exclusively on using a highly efficient vector representation as the input data and on the dataset size. Being a non-parametric method, it is often successful—as in this case—in classification situations where the decision boundary is very irregular. Nonetheless, in spite of its high accuracy and the minimal training phase, we favor the use of deep learning architectures for classification (such as CNN) over a kNN model because of its lower computational cost during prediction time. In fact, classifying a given observation requires a rundown of the whole dataset being too computationally expensive for large dataset as in this work.

All models were evaluated using a 5-fold cross-validation on incremental sample sizes (1K, 10K, 100K, 1M, samples) and on the largest available balanced set (using undersampling) with 4.01 million labeled samples from a total of 6 million. Overall, the models proved robust and had the ability to generalize. The best performance was achieved with kNN using a balanced sampling of the whole dataset. The results fit expectations, with an $F_1 = 0.93$ and an agreement of $\kappa = 0.84$. The fact that the CNN model outperformed FastText may be explained by its greater depth, but more efforts should be made to experiment with alternative CNN topologies.

This CNN model was also evaluated on an independent clinical data source, thus paving the way toward its potential use—taking into account pending improvements—in a clinical support system for oncologists when employing their clinical notes.

During the experiments, the word-embedding models achieved high-quality clusters, in addition to demonstrating their capacity for semantic reasoning, since they were able to identify the equivalent treatments for a type of tumor by means of an analogy with the drugs used to treat other tumors. These interesting reasoning qualities merit study in a future work using this dataset.

The evaluation results show that clinical trial protocols related to cancer, which are freely available, can be meaningfully exploited by applying representation learning, including deep learning techniques, thus opening up the potential to explore more ambitious goals by making the additional efforts required to build the appropriate dataset.

Our most immediate future work is to use a larger sample test of short clinical text from medical records for real simulation and include the effectiveness of CT interventions in the model, thus enabling us to not only predict whether or not a patient's case has been studied, but also whether the proposed treatment is expected to be effective based on the results of completed clinical trials for each indication. The problem would be a multilabel classification task, where the classes would be "effective" vs. "non-effective" and "studied" vs. "non-studied", and both could be either true or false. This would allow us to classify from four types of cases: effective and studied, potentially effective but not studied, not effective and studied, and potentially not effective and not studied. The main effort in this case lies in the dataset building, which entails including the obtained efficacy results for each study. As only a subset of CTs (5754 samples, 11%) have the results reported on clinicaltrials.gov, it means that, for this goal, it would be necessary to augment data from other sources, such as PubMed [38]. Following prior effort, a new model could be built to output potential cancer treatments that could be considered for a particular patient case based on the efficacy results of completed clinical trials.

Author Contributions: Conceptualization, A.B.; Investigation, A.B. and A.P.; Methodology, A.B. and A.P.; Software, A.B.; Formal Analysis, A.B. and A.P.; Data Curation, A.B.; Writing—Original Draft Preparation, A.B.; Writing—Review & Editing, A.P.; Visualization, A.B.; Supervision, A.P.; Funding Acquisition, A.P.

Funding: This work was supported by Medbravo, the Pattern Recognition and Artificial Intelligence Group (GRFIA) and the University Institute for Computing Research (IUII) from the University of Alicante.

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

References

1. National Comprehensive Cancer Network. *NCCN Clinical Practice Guidelines in Oncology*; National Comprehensive Cancer Network: Fort Washington, PA, USA, 2017.
2. Jin, S.; Pazdur, R.; Sridhara, R. Re-Evaluating Eligibility Criteria for Oncology Clinical Trials: Analysis of Investigational New Drug Applications in 2015. *J. Clin. Oncol.* **2017**, *35*, 3745–3752, doi:10.1200/JCO.2017.73.4186. [CrossRef] [PubMed]
3. Aronson, A.R. Effective mapping of biomedical text to the UMLS Metathesaurus: The MetaMap program. In Proceedings of the AMIA Symposium, Washington, DC, USA, 3–7 November 2001; p. 17.
4. Demner-Fushman, D.; Rogers, W.J.; Aronson, A.R. MetaMap Lite: An evaluation of a new Java implementation of MetaMap. *J. Am. Med. Inform. Assoc.* **2017**, *24*, 841–844. [CrossRef] [PubMed]
5. Savova, G.K.; Masanz, J.J.; Ogren, P.V.; Zheng, J.; Sohn, S.; Kipper-Schuler, K.C.; Chute, C.G. Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): Architecture, component evaluation and applications. *J. Am. Med. Inform. Assoc.* **2010**, *17*, 507–513. [CrossRef] [PubMed]
6. Leaman, R.; Islamaj Doğan, R.; Lu, Z. DNorm: Disease name normalization with pairwise learning to rank. *Bioinformatics* **2013**, *29*, 2909–2917. [CrossRef] [PubMed]
7. Savova, G.K.; Tseytlin, E.; Finan, S.; Castine, M.; Miller, T.; Medvedeva, O.; Harris, D.; Hochheiser, H.; Lin, C.; Chavan, G.; et al. DeepPhe: A Natural Language Processing System for Extracting Cancer Phenotypes from Clinical Records. *Cancer Res.* **2017**, *77*, e115–e118, doi:10.1158/0008-5472.CAN-17-0615. [CrossRef] [PubMed]
8. McEwan, R.; Melton, G.B.; Knoll, B.C.; Wang, Y.; Hultman, G.; Dale, J.L.; Meyer, T.; Pakhomov, S.V. NLP-PIER: A scalable natural language processing, indexing, and searching architecture for clinical notes. *AMIA Summits Transl. Sci. Proc.* **2016**, *2016*, 150–159. [PubMed]
9. Weng, C.; Tu, S.W.; Sim, I.; Richesson, R. Formal representation of eligibility criteria: A literature review. *J. Biomed. Informat.* **2010**, *43*, 451–467, doi:10.1016/j.jbi.2009.12.004. [CrossRef] [PubMed]
10. Huang, Z.; Ten Teije, A.; Van Harmelen, F. SemanticCT: A semantically-enabled system for clinical trials. In *Process Support and Knowledge Representation in Health Care*; Springer: Berlin, Germany, 2013; pp. 11–25.
11. Milian, K.; Hoekstra, R.; Bucur, A.; ten Teije, A.; van Harmelen, F.; Paulissen, J. Enhancing reuse of structured eligibility criteria and supporting their relaxation. *J. Biomed. Informat.* **2015**, *56*, 205–219, doi:10.1016/j.jbi.2015.05.005. [CrossRef] [PubMed]
12. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.

13. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444, doi:10.1038/nature14539. [CrossRef] [PubMed]
14. Mullenbach, J.; Wiegreffe, S.; Duke, J.; Sun, J.; Eisenstein, J. Explainable Prediction of Medical Codes from Clinical Text. *arXiv* **2018**, arXiv:1802.05695.
15. Hughes, M.; Li, I.; Kotoulas, S.; Suzumura, T. Medical Text Classification using Convolutional Neural Networks. *Stud. Health Technol. Inform.* **2017**, *235*, 246–250. [PubMed]
16. National Library of Medicine, National Institutes of Health. *XML Schema for ClinicalTrials.gov Public XML*; National Library of Medicine, National Institutes of Health: Bethesda, MD, USA, 2017.
17. Řehůřek, R.; Sojka, P. Software Framework for Topic Modelling with Large Corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks; ELRA: Valletta, Malta, 2010; pp. 45–50.
18. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26*; Burges, C.J.C., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2013; pp. 3111–3119.
19. Ling, C.X.; Sheng, V.S. Cost-sensitive Learning and the Class Imbalanced Problem. In *Encyclopedia of Machine Learning*; Sammut, C., Ed.; Springer: Berlin, Germany, 2007.
20. Bojanowski, P.; Grave, E.; Joulin, A.; Mikolov, T. Enriching Word Vectors with Subword Information. *arXiv* **2016**, arXiv:1607.04606.
21. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**, arXiv:1301.3781.
22. Bergstra, J.; Bengio, Y. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
23. Joulin, A.; Grave, E.; Bojanowski, P.; Mikolov, T. Bag of Tricks for Efficient Text Classification. *arXiv* **2016**, arXiv:1607.01759.
24. Kim, Y. Convolutional Neural Networks for Sentence Classification. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), Doha, Qatar, 25–29 October 2014; pp. 1746–1751.
25. Zhang, K.; Demner-Fushman, D. Automated classification of eligibility criteria in clinical trials to facilitate patient-trial matching for specific patient populations. *J. Am. Med. Inform. Assoc.* **2017**, *24*, 781–787, doi:10.1093/jamia/ocw176. [CrossRef] [PubMed]
26. Ni, Y.; Wright, J.; Perentesis, J.; Lingren, T.; Deleger, L.; Kaiser, M.; Kohane, I.; Solti, I. Increasing the efficiency of trial-patient matching: Automated clinical trial eligibility Pre-screening for pediatric oncology patients. *BMC Med. Inform. Decis. Mak.* **2015**, *15*, 28, doi:10.1186/s12911-015-0149-3. [CrossRef] [PubMed]
27. Das, A.; Thorbergosson, L.; Griogorenko, A.; Sontag, D.; Huerga, I. Using Machine Learning to Recommend Oncology Clinical Trials. Automated classification of eligibility criteria in clinical trials to facilitate patient-trial matching for specific patient populations. *MLHC Clin.* **2017**, doi:10.1186/s12911-015-0149-3. [CrossRef]
28. Landis, J.R.; Koch, G.G. The Measurement of Observer Agreement for Categorical Data. *Biometrics* **1977**, *33*, 159–174, doi:10.2307/2529310. [CrossRef] [PubMed]
29. Chollet, F. Keras. Available online: https://github.com/fchollet/keras (accessed on 22 July 2018).
30. Keras: Deep Learning for Humans. Available online: https://github.com/keras-team/keras (accessed on 22 July 2018).
31. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
32. McHugh, M.L. Interrater reliability: The kappa statistic. *Biochem. Med.* **2012**, *22*, 276–282. [CrossRef]
33. Keskar, N.S.; Mudigere, D.; Nocedal, J.; Smelyanskiy, M.; Tang, P.T.P. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. *arXiv* **2016**, arXiv:1609.04836.
34. Van Der Maaten, L.; Hinton, G. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
35. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *arXiv* **2016**, arXiv:1603.04467.
36. Macqueen, J. Some methods for classification and analysis of multivariate observations. In Proceedings of the 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, CA, USA, 1967; pp. 281–297. Available online: https://projecteuclid.org/euclid.bsmsp/1200512974 (accessed on 22 July 2018).

37. Levy, O.; Goldberg, Y. Linguistic regularities in sparse and explicit word representations. In Proceedings of the eighteenth Conference on Computational Natural Language Learning, Baltimore, MD, USA, 26–27 June 2014; pp. 171–180.
38. Pubmeddev. Home—PubMed—US National Library of Medicine National Institutes of Health (NCBI). Available online: https://www.ncbi.nlm.nih.gov/pubmed/ (accessed on 22 July 2018).