DL4papers: a deep learning approach for the automatic interpretation of scientific articles

L. Bugnon, C. Yones, J. Raad, M. Gerard,
M. Rubiolo, G. Merino, M. Pividori,
L. Di Persia, D.H. Milone and G. Stegmayer

Research Institute for Signals, Systems and Computational Intelligence, sinc(i),
FICH/UNL-CONICET, Ciudad Universitaria, (3000) Santa Fe, Argentina.

Abstract

Motivation: In precision medicine, next-generation sequencing and novel preclinical reports have led to an increasingly large amount of results, published in the scientific literature. However, identifying novel treatments or predicting a drug response in, for example, cancer patients, from the huge amount of papers available remains a laborious and challenging work. This task can be considered a text mining problem that requires reading a lot of academic documents for identifying a small set of papers describing specific relations between key terms. Due to the infeasibility of the manual curation of these relations, computational methods that can automatically identify them from the available literature are urgently needed.
Results: We present DL4papers, a new method based on deep learning that is capable of analyzing and interpreting papers in order to automatically extract relevant relations between specific keywords. DL4papers receives as input a query with the desired keywords, and it returns a ranked list of papers that contain meaningful associations between the keywords. The comparison against related methods showed that our proposal outperformed them in a cancer corpus. The reliability of the DL4papers output list was also measured, revealing that 100% of the first two documents retrieved for a particular search have relevant relations, in average. This shows that our model can guarantee that in the top-2 papers of the ranked list, the relation can be effectively found. Furthermore, the model is capable of highlighting, within each document, the specific fragments that have the associations of the input keywords. This can be very useful in order to pay attention only to the highlighted text, instead of reading the full paper. We believe that our proposal could be used as an accurate tool for rapidly identifying relationships between genes and their mutations, drug responses and treatments in the context of a certain disease. This new approach can certainly be a very useful and valuable resource for the advancement of the precision medicine field.
Availability and implementation: A web-demo is available at:
http://sinc.unl.edu.ar/web-demo/dl4papers/
Full source code and data are available at:
https://sourceforge.net/projects/sourcesinc/files/dl4papers/
Contact: lbugnon@sinc.unl.edu.ar

1 Introduction

Precision medicine is a growing field of research and, also, of technological development for improving human health and quality of life (Lin *et al.*, 2019). In precision medicine, it is assumed that the underlying molecular causes of disease are, in many cases, specific to each patient, thus identifying them can help to find the best treatment for each individual (Sboner and Elemento, 2016). In fact, personal sequencing data is very easy to have nowadays, quickly and at a low cost. The advantages of this approach, both for patients and physicians, include better and more accurate diagnosis and treatments, as well as safer drug prescription (Gomez-Lopez *et al.*, 2019). In such a scenario, physicians daily face lists of genomic alterations, where only a small minority of them could be relevant as biomarkers to drive clinical decision-making. For this reason, the medical community agrees on the urgent need of novel approaches for prioritizing treatments that could be clinically actionable in, for example, cancer therapy (Piñeiro-Yáñez *et al.*, 2018).

Selection of the right treatment to be delivered to the right patient and at the right time, based on the patient genomics profile, is the promise of precision medicine (Vanden Berghe and Hoste, 2019). To achieve this, it is important to identify specific molecular biomarkers, such as genes with a specific mutation, to predict drug efficacy on a patient. However, a certain biomarker can present a different response to the same gene inhibitor. For instance, it has been demonstrated that BRAF V600-mutated tumor types do not respond uniformly to BRAF-targeted therapy when treated with the drug Vemurafenib (Ducreux *et al.*, 2019). Thus, understanding the specific relation between a gene, its mutation and a drug, in a specific context (a given disease) can be crucial for the success of precision medicine (Lee *et al.*, 2018).

There are available databases containing gene-mutation-drug relations, built from curated literature on clinical studies (Levy *et al.*, 2011; Landrum *et al.*, 2016; Warner *et al.*, 2016). Unfortunately, manually curating (that is, doing text normalization and labeling relations) is a very difficult task to achieve due to the large number of on-going research projects, and the fast-growing volume of articles reporting new relations, all the time. To face it, many tools for the automatic recognition of individual biomedical entities (such as genes, mutations, drugs, diseases) in a text have been developed (Rocktaschel *et al.*, 2012; Wei *et al.*, 2013, 2015; Leaman *et al.*, 2015; Lee *et al.*, 2016c; Wang *et al.*, 2018). However, they are mainly focused on the identification of single words instead of finding meaningful relations among them, which remains a hard challenge that has not yet been effectively solved.

Several efforts in developing computational methods for the automatic extraction of relevant relations between biomedical entities from literature have been made. Some proposals can capture some specific relations between entities based on the co-occurrence of the words (Doughty *et al.*, 2011; Lee *et al.*, 2016b; Soto *et al.*, 2019), providing high sensitivity but low precision. As an alternative, machine learning (ML) approaches have been proposed more recently. For example, in (Singhal *et al.*, 2016), some features have been extracted such as the distance between each word and a keyword, or the frequencies of a "disease-word"' or "mutation-disease-word" co-occurrence. Then, a ML classifier for finding mutation-disease associations has been trained. However, identifying close relations between different entities (for example, gene-mutation or gene-drug) with high accuracy remains still a challenge. One of the most recent ML proposals consists of manually extracted features used together with a random forest classifier to identify keyword-pairs relationships in full documents (Lee *et al.*, 2018). Although this method has outperformed existing models, it requires a lot of hand-made curation and pre-processing steps to obtain a set of features for a collection of papers.

The fast-growing of deep learning (DL) algorithms (Bugnon *et al.*, 2019; Stegmayer *et al.*, 2018), together with the recent emergence of word embeddings (Mikolov *et al.*, 2013; Pennington *et al.*, 2014; Bojanowski *et al.*, 2017; Lee *et al.*, 2019) that represent a single word by a low-dimensional vector, capturing - in some way - its semantic information, have lead to substantial improvements in the field of automatic information extraction from text (Habibi *et al.*, 2017). In fact, text mining using DL has actually many advantages regarding automatic feature generation and extraction, which could lead to a better use of full documents. Deep neural networks can automatically extract the most relevant features from them, instead of the classical feature extraction techniques, which are extremely time-consuming and require the involvement of domain experts. Thus, a single DL model looking for relevant keyword-pair relations in the text could be the best way for analyzing full-text manuscripts.

Several DL models for finding relationships between biomedical entities have been recently proposed. For relation classification between two keywords with a representation at character-level, Li *et al.* (2017) proposed the combination of a convolutional neural network (CNN) with a long short-term memory (LSTM) for entity recognition, and another LSTM for sentence-level analysis. In (Peng and Lu, 2017) a CNN for predicting protein-protein interaction was proposed, with a feature that measures the distance between keywords in the same sentence. It was trained and evaluated at sentence-level. The authors state at the conclusions that it would be interesting to be able to extend the model beyond the sentence boundary. More recently, Peng *et al.* (2018) used an ensemble of support vector machines, CNNs and LSTMs, combined using either majority voting or stacking, in a chemical-protein relation corpus. Lee *et al.* (2018) proposed a model for sentence-level analysis based on CNNs and word embeddings pre-trained on Google News. All the deep learning models in the works cited above have been evaluated only for finding relations between pairs of keywords at a sentence-level. That is to say, not from a complete document, such as a PubMed manuscript.

A wider context at document-level, beyond just each single sentence, could certainly help a model to better find relationships. For example, in the case of a sentence having two keywords and located in the manuscript introduction, it can actually refer to a result that has not been found in the article under analysis, but in some previous works. In that case, the context could help to discern about the relevance of the relationship for the analyzed document. In fact, a sentence in the context of a results or conclusion section is more likely to be referring to the main topic of the article. Another example might be when a sentence includes the two input keywords with a positive relation, but then, in one of the following sentences the lack of statistical support for this result is indicated, refuting the positive relation. Also, the first sentences in a paragraph can say something like that a trial was made for one patient only, thus results are preliminary and not being conclusive regarding a complete population. Then, the results presented in the following sentences can confuse a sentence-level model. Finally, it should be taken into account that when two input keywords are not in the same sentence, the sentence is directly ignored by such model, precisely because of the lack of a wider context than the isolated sentence.

Here, we propose a novel approach for finding relationships among biomedical entities within complete texts, with a single stand-alone deep learning model and at document-level. Our tool, DL4papers, is capable of automatically extracting characteristics from a large pool of text, looking for relationships between user-defined entities. This means that it can find if there is a relation between, for example, a specific mutation and a drug, within all published papers on cancer in a repository. To achieve this, it receives as input the biomedical keywords of interest and all the papers in the repository where these keywords could be found. Then, each raw text is transformed into a numeric representation using a word embedding model. This data is used for feeding a deep learning network, which learns to extract useful text features

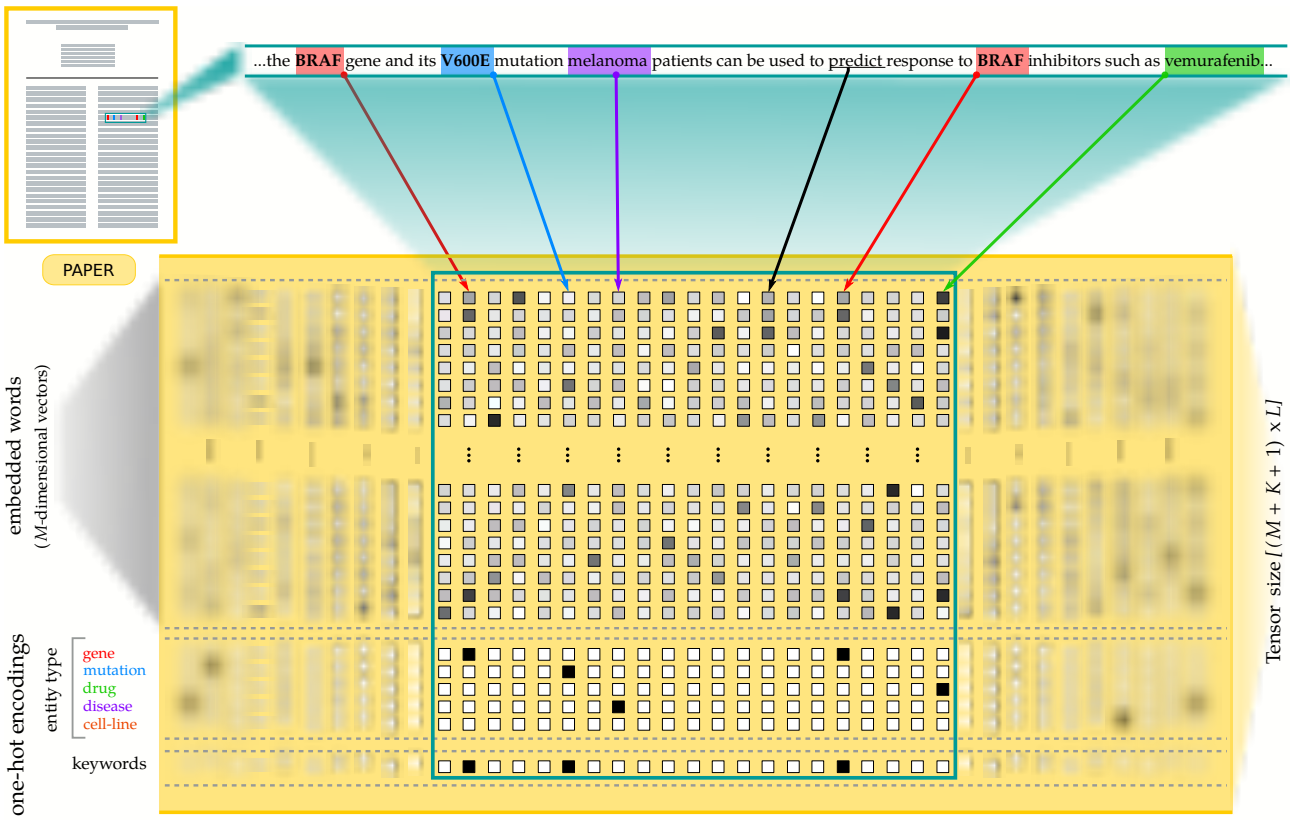

Figure 1: Word embedding designed for representing a document and a pair of input keywords (a specific example is shown for the gene "BRAF" and the mutation "V600E"). A document of L words is represented with a $M \times L$ embedding tensor. Additionally, two one-hot-encoding vectors for each word are defined. The entity-type vector of dimension K represents the category of the word (gene, mutation, drug, disease, cell-line). The 1-D keywords vector indicates which specific entity terms are in the input search query. Finally, all vectors are concatenated to form a tensor of $(M + K + 1) \times L$ elements as input to the DL model.

automatically. As the response, DL4papers returns a list of papers where relevant relations between the keywords are found. Furthermore, in order to better assist the process, the returned list of output papers is ordered, according to a relevance score, from more to less relevant, showing among the top-papers the ones where the strongest relations were identified. Besides, the fragments of texts that are the key ones for the relationships found are highlighted within the original text documents.

2 The deep learning model

DL4papers is based on an end-to-end DL model. It is trained with full documents and several keyword-pairs with binary labels, indicating whether there is, or there is not, a relation between them within each text. Thus, given the input keywords, the model outputs a prediction score for each manuscript in the corpus, indicating the probability that the input keywords are related in the analyzed document.

For training the DL model, a corpus of documents is vectorized using a word embedding. It takes the text corpus and produces a vector space of M dimensions in which each word in the corpus is assigned to a point (embedded vector) in the space (see Figure 1). In order to represent a full text of L words, all the word vectors in the text are concatenated to form a tensor of $M \times L$ elements. Then, two types of one-hot-encoding vectors for each word in the text are generated. The first one is the entity-type encoding that indicates to which of the possible K entities the word belongs to (if any). For example, the entities could be gene, drug, mutation, disease or cell-line ($K = 5$). The second one is the keyword encoding that indicates which specific keywords are of interest and related in the text (1 if the word is part of the keywords-pair and 0 in the other case). These two encodings form a $K + 1$ vector for each word. It is important to note that only the keyword encoding part is modified for each specific user search, thus the heaviest part of the processing must be run only once. An example of the resulting tensor is shown in Figure 1, with a specific example for the BRAF gene and its V600E mutation. The vector generated by the word embedding, the entity-type one-hot-encoding vector and the keyword one-hot-encoded vector are concatenated, to form a $(M+K+1) \times L$ tensor, for each part of the document in the corpus. It should be noted that this embedding design has the potential of allowing the input of several biomedical entities at the same time.

The full end-to-end architecture of the DL model is shown in Figure 2. Given an input keyword pair,

the tensor that represents a document first passes through two convolutional layers, whose main function is to compress the word embeddings from M to a smaller dimension N. Then, the resulting $N \times L$ tensor goes through a series of convolutional layers. The number of convolutional layers in the network is an important parameter that can lead to a degradation of the accuracy of the network, as reported in (He *et al.*, 2016a). Therefore, the convolutional layers are grouped in a series of fully pre-activation identity blocks, as suggested in (He *et al.*, 2016b). These blocks, also known as residual and pooling layers, propagate the input through the whole network with identity operations. Each residual layer is composed of two convolutional layers, with their corresponding ELU activations (Clevert *et al.*, 2016) and batch normalization layers (Ioffe and Szegedy, 2015). This results in neural networks with a better generalization capability for the deepest configurations.

Phrases that relate the input keywords could be located in any section of the document, many representative examples of these situations are needed for training a model. One possible solution is to use data augmentation, but differently from images where it is widely used, a 1-dimensional translation of the text could break important semantic relations. Therefore a global max pooling was used in the last layer instead of a standard fully connected layer. This particular design of the last layer allows the first layers to be trained independently of the keywords position in the text, given that the parameters of the first layers can be learnt from sentences that are anywhere in the document. Besides, the global max pooling of the output layer, differently from most commonly used fully connected layer, has less weights to fit for the same performance, and it has been demonstrated that this type of layer has less overfitting problems (Lin *et al.*, 2013). Moreover, it provides a more interpretable output for the model because the position of the maximum is related to the location of the most activated feature in the input.

For many layers, the number of filters in each layer might generate a large number of combinations of hyperparameters to be tuned, which can be a challenge to achieve in a reasonable amount of time. Therefore, the network architecture was designed in order to have a small number of hyperparameters. For example, the number of layers has a direct impact on the receptive field of the network. If only one convolutional layer with a filter size of three is used, each element of the activation map depends on, only, three words. Since the final prediction of our model is the maximum value of the activation map, it would depend only on a portion composed by three words of the text. As shown in (He *et al.*, 2016b), an interesting property of the identity blocks is that a network with more layers results in equal or better performance. This happens because the shortcut connections allow the network to easily ignore layers that are not needed. The identity blocks allow the model to auto-define the number of convolutional layers needed during training, avoiding the optimization of the number of hidden layers. When there are many identity blocks available in the architecture, the model is capable of automatically selecting how many of these blocks are really necessary . The non-necessary blocks will simply be skipped. This means that there is no need to look for the optimum number of blocks (He *et al.*, 2016b). For this reason, in our model we have used several (R) identity blocks (and their corresponding average layer) with the minimum kernel size (3). Larger filters are not needed, since the large number of identity blocks results in a large enough receptive field. Furthermore, in (He *et al.*, 2016b) it was shown that the main path of the network must be unobstructed to correctly backpropagate the errors. Therefore, to avoid convolutional layers to interrupt the main path of the network, the number of filters used on each convolutional layer of each identity block was the same, except for the first layer that had twice as many filters than the other ones; and for the last layer, which only had one filter.

In summary, the architecture of the DL4papers model has several differences and advantages with respect to previous deep learning models. First of all, it is capable of processing full texts as inputs. It has been designed to work at document-level representation, differently from the previous works presented in the introduction, which can work only at the sentence-level. Besides, those models are combined with other classical ML models, such as RF or ensembles. Our proposal, instead, is a completely end-to-end deep learning model, providing a ranked list of articles from more to less relevant to the input keywords. As detailed above, the model has identity blocks that avoid having to look for an optimum number of layers, and the global max pooling provides more interpretability to the model results.

3 Materials and methods

3.1 Data

For evaluation of the proposed DL model and fair comparison with related methods on the same public and freely available data, we have used the recently developed BRONCO[1] dataset, which is a curated corpus including biomedical entities in oncology (Lee *et al.*, 2016a). According to their authors, it is currently the largest full-text variant-centric corpus annotated with related genes and its mutations, diseases, drugs and cell-line information. It contains 108 full-text articles, whereas other corpora from previous studies contain only abstracts or isolated sentences. Its articles are related to cancer and anti-tumor drug screening research, and they were manually curated by biomedical experts. BRONCO has information on 403 mutations and their associations with genes, diseases and drugs. Keeping in mind the possible keyword-pairs, it is worth mentioning that in this dataset each mutation has only one associated gene (Lee *et al.*, 2018). The mutation-gene search will be done only for a fair comparison with related published methods.

[1] http://infos.korea.ac.kr/bronco/

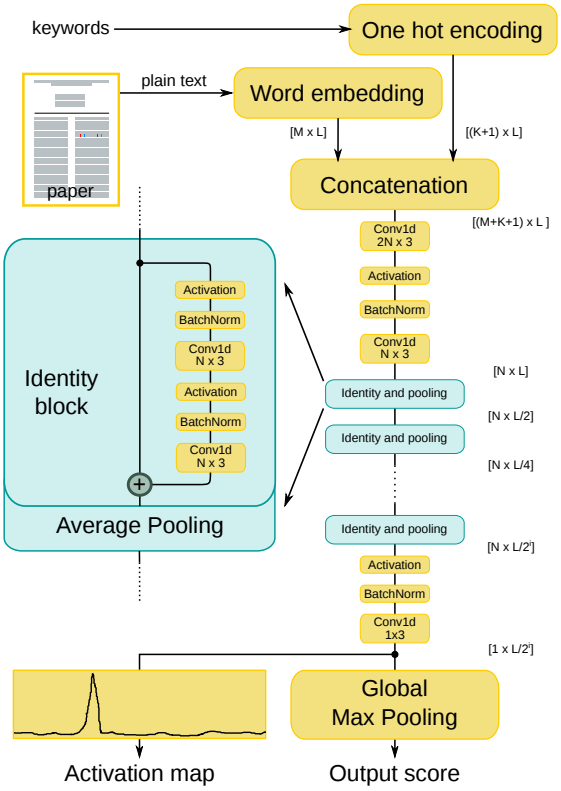

Figure 2: Schematic representation of DL4papers end-to-end architecture.

3.2 Experimental setup

For each full document, the mutation keywords are already available in BRONCO. The other keywords (genes, drugs, diseases and cell-lines) were retrieved from the texts using the dictionary *Biomedical Entity Extractor*[2] as in (Lee *et al.*, 2018). Then, for each text, all possible mapping pairs among the biomedical entities were obtained. Two different types of cases were evaluated: mutation-gene and mutation-drug. Thus, given all the mutations, genes and drugs that appear in the same text, all possible candidate mutation-gene and mutation-drug relation pairs were identified. In this document-level extraction, even though the two entities were not in the same sentence, the relations remained in the candidate set. For example, when the total number of unique mutations in a document is w_1, and the total number of drugs (or genes) is w_2, the possible relations are $w_1 \times w_2$. Since the goal was building a DL model that classifies these relations as true or false, an input pair was considered true if it appeared in the list of positive relations defined in BRONCO, and false otherwise. Among all the relations defined in BRONCO, 285 mutation-gene tagged as true (positive relationship), and 11,641 as false (negative relationship) were found, which makes a class imbalance of 1:40. Whereas 209 true and 3,285 false mutation-drug relations have been identified, with a class imbalance of 1:10. It should be mentioned that only 115 out of these 209 positive cases have both keywords in the same sentence. Thus, any kind of sentence-level model for the mutation-drug case would have, at the best, 45% of sensitivity for document classification in this dataset.

For the DL model, a 10-fold cross-validation has been done. In each fold, we have used independent training and testing partitions and the model parameters (weights) have been tuned with stochastic gradient descent. The same hyperparameters have been used in each fold ($R = 12, N = 6$, filter size $= 3$). The dimension $K = 5$ of the one-hot-encoding for entity-type was used because there are five possible entities to be found in this dataset (gene, mutation, drug, disease or cell-line). The embedding dimension M was defined by the embedding used. The length L was set to 2^{14} considering the longest document available in the corpus. Shorter texts than L can be safely zero-padded to fill the complete tensor, since the network does not have any fully connected layer at the output, thus it is invariant to translation. Regarding the identity blocks, since they can be used (or skipped) by the deep network model for better modeling the training data, the number of identity blocks is not a sensitive parameter (see Supplementary Material). That is, if many identity blocks are available in the architecture, the model is capable of automatically selecting by itself how many of these blocks are really necessary for better training. In general, it can be stated that the model architecture is robust to a wide range of hyperparameters, as it is shown in the sensitivity analysis presented in the Supplementary Material.

As it was previously depicted, there is a substantial imbalance between the number of positive and negative data points in the set of relation candidates. This is a clear example of a realistic scenario,

[2]http://infos.korea.ac.kr/bioentityextractor/

where it is expected that the tool should retrieve just a few (the most) relevant articles among a lot of full-text manuscripts, such as for example, all PubMed database. Thus, 10-fold cross-validation for both, mutation-gene and mutation-drug cases considering the natural class imbalance existing in BRONCO were done. Then, a second experiment also involving a 10-fold cross validation was performed for the comparison with state-of-the-art methods. To fairly compare our model with the ones reported in [3], where balanced training and testing sets were used, negative cases were randomly sampled to match the positive ones as in (Lee *et al.*, 2018).

3.3 Performance evaluation

The methods performance is reported with standard evaluation metrics,

$$\begin{aligned} p &= \frac{TP}{TP+FP}, \quad s^+ = \frac{TP}{TP+FN}, \quad F_1 = 2\frac{s^+ p}{s^+ + p}, \\ MCC &= \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}, \\ G_m &= \sqrt{s^+ p}, \end{aligned}$$

where TP, TN, FP and FN are the number of true positives, true negatives, false positives and false negatives, respectively. The sensitivity (s^+) measures how good a classification method is for recognizing (and not missing) the TPs of the problem. The precision (p) measures the relation between TPs and FPs. In a realistic scenario for practical applications, precision is very important in imbalanced datasets because FPs can be many more than the TPs. Thus, considering the characteristics of the classification problem under study, it is important to take into account both sensitivity and precision. Therefore, F_1 is used as a global comparative measure, together with Matthew correlation coefficient (MCC) and G_m, the geometric mean of sensitivity and precision, which are particularly used for imbalanced datasets.

In a first stage, DL4papers was evaluated using the original class imbalance of the dataset. In this case, performance measures achieved by DL4papers using different embeddings was evaluated. Three embedding variants were compared: FastText (Bojanowski *et al.*, 2017), Word2Vec (Mikolov *et al.*, 2013) and GloVe (Pennington *et al.*, 2014), by means of the Wilcoxon test for paired samples and considering a significance threshold of $\alpha = 0.05$. The standard pre-trained word embedding of each variant has been used as provided by the original authors. This is a common practice that has shown to be better than training the embedding for a specific domain (Lee *et al.*, 2018).

For the second evaluation, the source code and data partitions used in (Lee *et al.*, 2018) are not publicly available. In addition, performance measures have been only reported in terms of average scores without providing standard deviations, interquartile ranges nor minimum and maximum achieved along the folds. Thus, the results achieved by DL4papers were contrasted with state-of-art methods average results using one-sample Wilcoxon signed-rank tests and considering a significance threshold of 0.05. In both scenarios, standard deviations among the ten folds and for all performance measures were also computed.

In addition, in a real case DL4papers will receive an input keywords pair and will return a list of papers, sorted by relevance score. Therefore, it is highly desirable that the first papers contain the strongest relations between the keyword terms. Thus, in order to evaluate the tool performance in such truly realistic scenario, we have measured the top-1, top-2 and top-3 recall of the output list of papers. That is, for all possible pairs of input queries in BRONCO, the top-n recall considers a hit if at least one paper within the first n articles in the ordered list has a true relation. The average of top-n recalls, per fold and per input keywords pair, is the measure reported in Section 4.3.

4 Results and discussion

4.1 Evaluation of embeddings

Figure 3 shows the performance of DL4papers when using three possible embedding variants: FastText, Word2Vec and GloVe. The differences among the embedding variants lies on the fact that Word2Vec can be considered as a very large dictionary of existing and generic English terms. The dimension of the embedding is 300 and its main assumption is that semantic analogies can be preserved under basic arithmetics on the word vectors, which are trained to be predictive of analogies within a specific domain. In this approach, the distance between two words is fundamental for correct domain representation. FastText is based on character n-grams. Its embedding dimension is also 300 but it can generate valid embeddings for any word, which makes it more suitable for technical documents. GloVe considers the corpus word occurrence statistics for representation. It tries to achieve analogy preservation under linear arithmetics of the statistical properties of the corpus, which are those actually used as inputs. The dimension of this embedding is 100. The left panels in Figure 3 present the boxplots of the performance measures achieved across 10 folds when looking for mutation-gene relations, whereas the right panels show the same results but for mutation-drug.

As it can be seen in the boxplot, in the case of the mutation-gene relations (left panels) all performance values are higher than 0.7, except for GloVe precision. The median sensitivity were 0.828 and 0.759, and

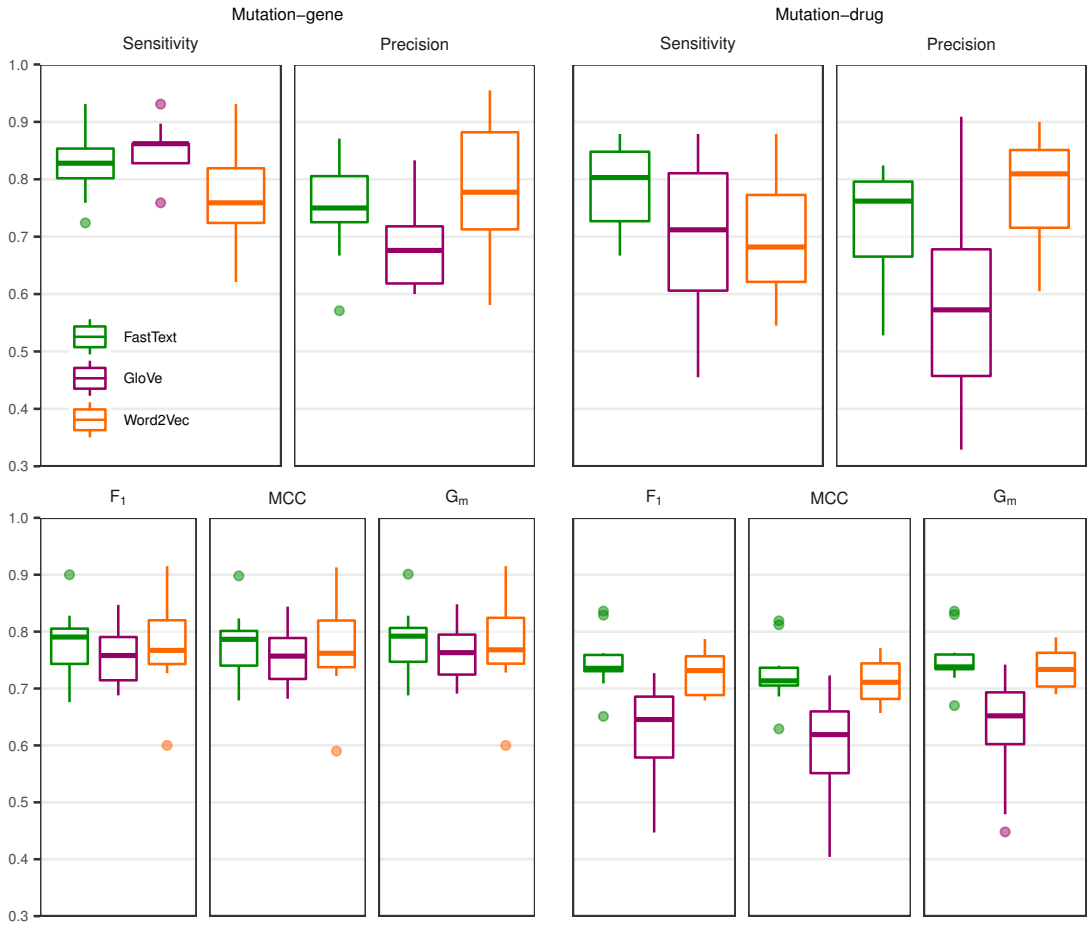

Figure 3: Performance measures: sensitivity (s^+), precision (p), F_1, MCC and G_m of DL4papers using three embedding variants: FastText, Word2Vec and GloVe. Models trained and tested with unbalanced data, for mutation-gene (left) and mutation-drug (right) relations from full documents in BRONCO.

precisions were 0.750 and 0.778 for FastText and Word2Vec, respectively. In the case of GloVe, in spite of having a high sensitivity (0.862), it reached the lowest values for precision (0.683). Regarding F_1, MCC and G_m, it can be clearly seen that they all three provided very similar scores. The statistical analysis of sensitivity and precision (one-tail two-paired-samples Wilcoxon test with $\alpha = 0.05$), has shown that FastText was more sensitive than Word2Vec and more precise than GloVe.

For the mutation-drug case (right panel), median values were higher than 0.7 in all cases and for all embeddings, except for GloVe in precision (0.572) and Word2Vec in sensivitity (0.682). As in the previous case, GloVe has shown the worst performance in most indexes. Median sensitivity was 0.803 and 0.682; precision was 0.762 and 0.810 using FastText and Word2Vec, respectively. GloVe is the method with higher variance and worst results. It is also significatively worse than the other methods. Regarding F_1, MCC and G_m, similarly to the previous case all the performance measures provided very similar scores. FasText was better than Word2Vec in sensitivity (one-tail two-paired-samples Wilcoxon test with $\alpha = 0.05$). Except for the comparison of sensitivity against Word2Vec, Glove scores were significantly lower than those achieved by the other embeddings (one-tail two-paired-samples Wilcoxon test with $\alpha = 0.05$). Therefore, DL4papers and FastText has shown to be the best configuration.

In summary, due to its low performance in all cases, GloVe has been discarded for the next experiments. The difference in the performance of this method in comparison to the other ones can be mainly due to its lower dimension. The results show that it was not capable of correctly capturing the context of the input keywords at the required document-level context. Moreover, GloVe takes into account word to word co-occurrence, thus it does not work well if many specific words are not part of the training corpus. Due to these best performance results, FastText was used as the word-embedding method of DL4papers in the next sections. Regarding performance metrics, since F_1, MCC and G_m have provided similar results, from now on only F_1 will be used in the following sections.

Table 1: Average top-n recall for DL4papers. The top-n recall for a particular keyword pair is 100% if there is at least one true positive document among the first n articles returned by DL4papers.

Keywords	top-1 recall	top-2 recall	top-3 recall
mutation-gene	91%	100%	100%
mutation-drug	83%	100%	100%

4.2 Comparison with related methods

Figure 4 shows the results of the comparison of DL4papers with other state-of-the-art methods for the same task of extracting relations between medical entities in a corpus of full texts. The boxplots represent the performance measures achieved across 10 folds by DL4papers, and the circle and triangle dots are the average scores for methods in Lee *et al.* (2018) and Singhal *et al.* (2016), respectively. Results of both methods for BRONCO dataset are those originally reported in (Lee *et al.*, 2018).

It can be observed that all methods achieved very high performance scores for the mutation-gene relation (left panel), mostly higher than 0.900. In particular, Lee et al. and Singhal et al. achieved 0.958 and 0.880 in sensitivity, 0.961 and 0.958 in precision, 0.958 and 0.913 in F1, respectively. Whereas, the scores of DL4 papers were 0.955, 0.948 and 0.951, for sensitivity, precision and F_1, respectively. The performance of DL4papers and the two alternative methods was not statistically different regarding precision (one-sample Wilcoxon test p-value=0.759 for both comparisons). In addition, no significant differences were found between DL4papers and Lee et al. recalls and F_1 (one-sample Wilcoxon test p-value equal to 0.837 and 0.683, respectively). On the other hand, DL4papers was significantly better than (Singhal *et al.*, 2016) for these two measures (one-tailed, one-sample Wilcoxon test p-value = 0.004 and 0.02, respectively). These results are shown in the left panel of Figure 4, where sensitivity and F_1 for the other methods fall within DL4papers boxplots. As it was stated before, in the BRONCO dataset each mutation has only one associated gene and they are usually mentioned together in an article. Thus, looking for a mutation-gene relation in this database is a simple task, and moreover when the testing and training datasets are balanced. However, it should be mentioned that in BRONCO there are some gene-mutation relations that appear only once, and therefore in only one document within the complete corpus. During model training with random folds, if this relationship appears only in a test partition, none of the models will be able to learn it. And thus, average sensitivity and precision cannot be 100%. Thus, this fact explains the results achieved by the three compared methods.

In the case of the mutation-drug relations (right panel), performance of DL4papers was significantly higher than those achieved for the two comparative methods (one-tailed, one-sample Wilcoxon test p-values ¡ 0.05 for all comparisons). As it can be seen, all comparative approaches are outside and below the corresponding DL4papers boxes. In particular, none of the other methods achieved more than 0.830 in the measures. DL4papers, instead, achieved the highest performance in all measures, and with standard deviations that do not include the other methods. Specifically, DL4papers reached 0.915(+/- 0.0711) for sensitivity, 0.935(+/- 0.0312) for precision, and 0.922 (+/- 0.0276) for F_1.

For all the methods, the difference in performance for both cases (mutation-drug and mutation-gene) can be explained by the fact that within BRONCO there are one-to-many (1:n) mutation-drug relations. This scenario is, without any doubt, a much challenging and closer-to-real-life problem than the one-to-one (1:1) relations in the mutation-gene case. It has to be highlighted that, unlike the other methods, which did not perform well for the more challenging mutation-drug task, DL4papers was able to achieve a very good performance.

Comparing the scores of DL4papers using FastText as the word-embedding tool, shown in Figure 3, with those in Figure 4, it can be noticed that, for both cases, the latter were much higher. This is due to the fact that in the first case a more realistic scenario was considered, with a large imbalance between classes in both training and testing datasets. However, as mentioned before, in the second case the datasets were balanced for a fair comparison with the competitor methods. It is interesting to note that although class imbalance here was 1:40 in the testing dataset (Figure 3), with more than 10,000 available cases and only around 300 positives, DL4papers reached a F_1 of around 75%.

4.3 DL4papers findings in a real case-study

The average sensitivity, measuring the reliability of DL4papers findings in a real and practical case is reported in Table 1 as top-1, top-2 and top-3 recall. Particularly, these results were obtained considering the keyword-pairs that had appeared in at least two papers (true positives). For the mutation-gene case, it was found that the highest output score was always assigned to a relevant paper at the top-2 and top-3 recall. That is why the average top-n recall for DL4papers was 91% only for the top-1, and 100% for all the other cases evaluated. Similarly, for the mutation-drug case and the top-1 recall, just a few cases of the output list were lost. However, in those cases, the true positive document appeared in the second position of the ordered output list. For this reason, the average top-1 recall was 83% but the top-2 and the top-3 recall were 100%. In summary, the top-n results show that, given a specific keywords pair, the output list

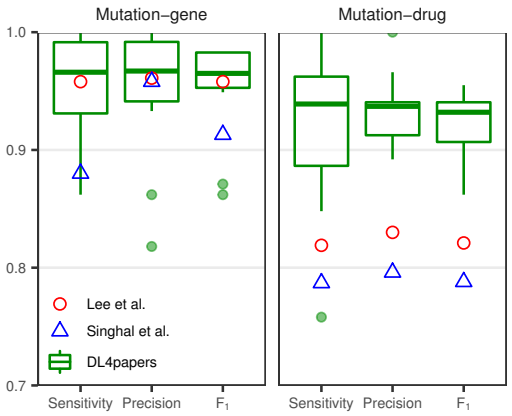

Figure 4: Performance scores in cross-validation and balanced experiments for mutation-gene (left panel) and mutation-drug (right panel) relations. Performance measures from state-of-the art methods are indicated for comparison.

generated by DL4papers is highly reliable. That is, it can be assured that only the top-2 articles have to be read, because we can assure that a true relationship between the keywords can be found in them.

Some examples of concrete cases and their corresponding outputs, with mutation-gene and mutation-drug keywords, were further explored. DL4papers was set to highlight the regions in the original text of an article where the evidence of relations between the input keywords were found. This was achieved by looking at the activation values of the last layers of the model for a particular output. By backpropagating the output signal towards the input layer, it is possible to highlight the original text in the input text that correspond to the particular output analyzed. We have used class activation mapping (CAM) (Zhou *et al.*, 2016) at the output activation map and calculated backwards the influence zone of each high value, using the filter sizes and the number of convolutive layers. Given the number of layers and filters sizes for each layer, the receptive field that the network used to generate the activation map peaks can be located and used to highlight the corresponding text. This is a very desirable characteristic for a model, which certainly contributes to the explainability of the results (Rudin, 2019). Figure 5a) shows an example of one of the results for the case of the pair mutation-gene V600E and BRAF. As it can be seen, the specific parts of the text indicated by the deep model explicitly show the relationship between BRAF and its mutation. This reveals why this paper resulted to be a positive response to the V600E-BRAF relation. In Figure 5b) an analogous case is shown for the specific mutation-drug, R175H and NSC319726 input keywords. In the Supplementary Material, two other real cases are analyzed in more detail.

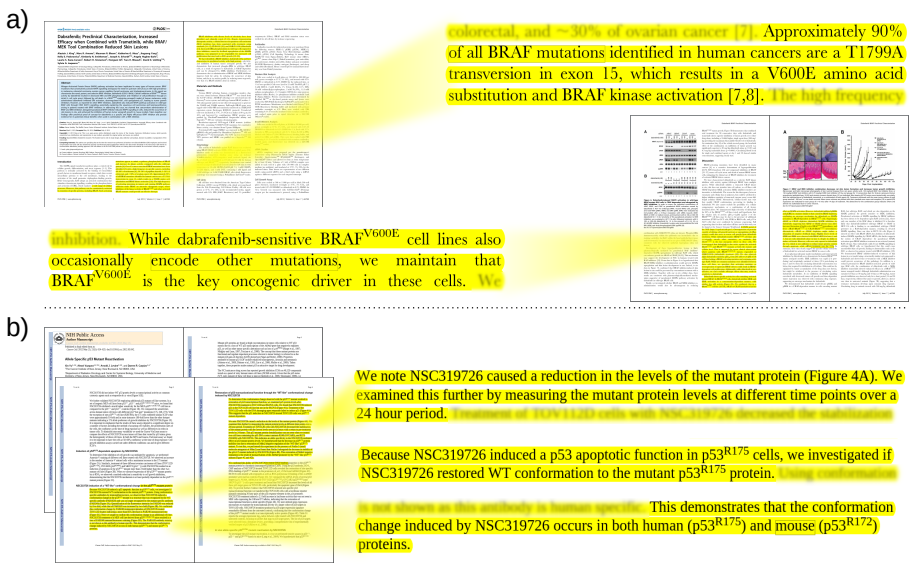

Figure 5: Examples of some possible outputs of DL4papers for: (a) mutation-gene output for V600E and BRAF keywords; and (b) mutation-drug output for R175 and NSC319726 keywords.

5 Conclusions

In this work a new method capable of automatically extracting relations between specific keywords and from full documents, DL4papers, was presented. The model receives as input a keywords pair and a corpus of articles, and it returns a ranked list of the documents containing relevant relations between those keywords. The core of DL4papers is a deep convolutional neural network that receives as input a word embedding of the article. Our proposal outperformed state-of-the-art methods for mutation-gene and mutation-drug relations in a publicly available corpus of cancer research. Furthermore, the high reliability of DL4papers results was demonstrated revealing that articles with true relations are always returned at the first or the second position of the output ranked documents. Our results indicate that DL4papers is a powerful tool that can help the advancing of many research and professional areas where keyword relations in full documents need to be discovered. Regardless it has been evaluated with an oncology corpus, the approach could be fully replicated and reused in any other application domain, as well. Moreover, the architecture behind DL4papers allows the input not only of two keywords, but also of any number and types of keywords. This is a very valuable feature of our tool. For instance, it could assist in the accurate and fast identification of the most relevant relationships among genes, mutations, drugs responses, diseases, cell-lines and treatments from literature.

Acknowledgments

We acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. This work was supported by ANPCyT (PICT 2014 #2627; 2018 #3384) and UNL (CAI+D 2016 #082). *Conflict of Interest:* none declared.

References

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, **5**, 135–146.

Bugnon, L. A., Yones, C., Milone, D. H., and Stegmayer, G. (2019). Deep neural architectures for highly imbalanced data in bioinformatics. *IEEE Transactions on Neural Networks and Learning Systems*, **5**(1), 1–10.

Clevert, D., Unterthiner, T., and Hochreiter, S. (2016). Fast and accurate deep network learning by exponential linear units (elus). In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Doughty, E., Kertesz-Farkas, A., Bodenreider, O., and et al. (2011). Toward an automatic method for extracting cancer- and other disease-related point mutations from the biomedical literature. *Bioinformatics*, **27**(3), 408–415.

Ducreux, M., Chamseddine, A., Laurent-Puig, P., and et al. (2019). Molecular targeted therapy of BRAF-mutant colorectal cancer. *Therapeutic Advances in Medical Oncology*, **11**, 1–15.

Gomez-Lopez, G., Dopazo, J., Cigudosa, J. C., Valencia, A., and Al-Shahrour, F. (2019). Precision medicine needs pioneering clinical bioinformaticians. *Briefings in Bioinformatics*, **20**(3), 752–766.

Habibi, M., Weber, L., Neves, M., Wiegandt, D. L., and Leser, U. (2017). Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, **33**(14), i37–i48.

He, K., Zhang, X., Ren, S., and Sun, J. (2016a). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

He, K., Zhang, X., Ren, S., and Sun, J. (2016b). Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 448–456.

Landrum, M. J., Lee, J. M., Benson, M., and et al. (2016). ClinVar: public archive of interpretations of clinically relevant variants. *Nucleic Acids Research*, **44**(D1), D862–D868.

Leaman, R., Wei, C.-H., and Lu, Z. (2015). tmchem: a high performance approach for chemical named entity recognition and normalization. *Journal of Cheminformatics*, **7**(1), S3.

Lee, J., Yoon, W., Kim, S., and et al. (2019). Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, **1**(1), 1–10.

Lee, K., Lee, S., Park, S., and et al. (2016a). BRONCO: Biomedical entity Relation ONcology COrpus for extracting gene-variant-disease-drug relations. *Database*, **2016**.

Lee, K., Shin, W., Kim, B., and et al. (2016b). HiPub: translating PubMed and PMC texts to networks for knowledge discovery. *Bioinformatics*, **32**(18), 2886–2888.

Lee, K., Kim, B., Choi, Y., and et al. (2018). Deep learning of mutation-gene-drug relations from the literature. *BMC Bioinformatics*, **19**(1), 21.

Lee, S., Kim, D., Lee, K., and et al. (2016c). BEST: Next-generation biomedical entity search tool for knowledge discovery from biomedical literature. *PloS one*, **11**(10), e0164680.

Levy, M. A., Lovly, C. M., Horn, L., and et al. (2011). My cancer genome: Web-based clinical decision support for genome-directed lung cancer treatment. *Journal of Clinical Oncology*, **29**(15_suppl), 7576–7576.

Li, F., Zhang, M., Fu, G., and Ji, D. (2017). A neural joint model for entity and relation extraction from biomedical text. *BMC bioinformatics*, **18**(198), 1–11.

Lin, C.-H., Konecki, D. M., Lichtarge, O., and et al. (2019). Multimodal network diffusion predicts future diseasegenechemical associations. *Bioinformatics*, **35**(9), 1536–1543.

Lin, M., Chen, Q., and Yan, S. (2013). Network in network. *CoRR*, **abs/1312.4400**, 1–10.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. *coRR*, **abs/1301.3781**, 1–10.

Peng, Y. and Lu, Z. (2017). Deep learning for extracting protein-protein interactions from biomedical literature. *Proceedings of the BioNLP 2017 workshop. Association for Computational Linguistics*, **1**(1), 29–38.

Peng, Y., Rios, A., Kavuluru, R., and Lu, Z. (2018). Extracting chemicalprotein relations with ensembles of SVM and deep learning models. *Database*, **1**(1), 1–10.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, volume 1, pages 1532–1543.

Piñeiro-Yáñez, E., Reboiro-Jato, M., Gómez-López, and et al. (2018). Pandrugs: a novel method to prioritize anticancer drug treatments according to individual genomic data. *Genome Medicine*, **10**(1), 41.

Rocktaschel, T., Weidlich, M., and Leser, U. (2012). ChemSpot: a hybrid system for chemical named entity recognition. *Bioinformatics*, **28**(12), 1633–1640.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, **1**, 206–215.

Sboner, A. and Elemento, O. (2016). A primer on precision medicine informatics. *Briefings in Bioinformatics*, **17**(1), 145–153.

Singhal, A., Simmons, M., and Lu, Z. (2016). Text mining for precision medicine: automating disease-mutation relationship extraction from biomedical literature. *Journal of the American Medical Informatics Association*, **23**(4), 766–772.

Soto, A. J., Przybya, P., and Ananiadou, S. (2019). Thalia: semantic search engine for biomedical abstracts. *Bioinformatics*, **35**(10), 1799–1801.

Stegmayer, G., Di Persia, L. E., Rubiolo, M., and et al. (2018). Predicting novel microRNA: a comprehensive comparison of machine learning approaches. *Briefings in Bioinformatics*, **5**, 1–15.

Vanden Berghe, T. and Hoste, E. (2019). Paving the way for precision medicine v2.0 in intensive care by profiling necroinflammation in biofluids. *Cell Death and Differentiation*, **26**(1), 83–98.

Wang, X., Zhang, Y., Ren, X., and et al. (2018). Cross-type biomedical named entity recognition with deep multi-task learning. *Bioinformatics*, **35**(10), 1745–1752.

Warner, J. L., Jain, S. K., and Levy, M. A. (2016). Integrating cancer genomic data into electronic health records. *Genome Medicine*, **8**(1), 113.

Wei, C.-H., Harris, B. R., Kao, H.-Y., and Lu, Z. (2013). tmVar: a text mining approach for extracting sequence variants in biomedical literature. *Bioinformatics*, **29**(11), 1433–1439.

Wei, C.-H., Kao, H.-Y., and Lu, Z. (2015). GNormPlus: An Integrative Approach for Tagging Genes, Gene Families, and Protein Domains. *BioMed Research International*, **1**(918710), 1–7.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. (2016). Learning deep features for discriminative localization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 1–6.