Extraction of Drug-Drug Interactions from Biomedical Texts

Albert Espín
Lavanya Mandadapu

Abstract

Two common challenges in the field of Natural Language Processing applied to biomedical texts are the recognition and classification of Drug Name Entities and the detection and classification of Drug-Drug Interactions (DDI). A specific competition to solve these two problems was proposed in SemEval 2013 Task 9. This work explains new models developed to approach these tasks, starting with baseline rule-based algorithms that are outperformed by later machine learning systems. A CRF model is used for Drug Name Entity recognition and classification. This model integrates features such as word shapes and clusters of word embeddings, as proposed by Liu et al. [1]. A macro-average test F1-score of 0.74 is obtained in the setting of the competition, significantly higher than the original winner system, that scored 0.64. An Ensemble of Deep Neural Networks is used to solve the DDI detection and classification task. The Ensemble is composed by 8 models that make individual predictions and vote to establish the final decision. These models include Convolutional Neural Networks (partially following the model of Liu et al. [2]), Recurrent Neural Networks and hybrid models. The Ensemble obtains a test F1-score of 0.63, only 2% lower than the winner of original task, that used Support Vector Machines.

1 Introduction

This work is devoted to explain the experimental process used to solve the problems proposed in SemEval 2013 Task 9 competition [3]: the recognition and classification of drug names (Task 9.1) and the extraction and classification of drug-drug interactions (Task 9.2).

The data to use is the DDI corpus [4], a set of XML files containing sentences where the names and types of drugs as well as their interactions are annotated, indicating their start and end positions in each sentence. The data has two initial partitions: training and test. The two challenges are solved developing machine learning models capable of learning from a set of features extracted from the training data along with the annotated targets, and evaluate the results on the test data, for which features are also extracted. The evaluation is given in terms of the F1 measure, which combines the results of precision and recall.

To select both the best features subsets and the best parameters of the learning algorithm, part of the training data (10%) is used for the validation of the results of different feature and parameter combinations tested on the remaining part of the data (90%). The best combinations are afterwards used to train the model with all the training data and assess the performance on the test set. This way, instead of optimizing the configuration directly on the test set, generalization is preserved, i.e. the learned model would be able to perform better in new hypothetical test data sets.

2 Task 9.1: Recognition and Classification of Drug Names

2.1 Baseline rule-based model

This trivial model is based on the first laboratory session material of the AHLT course. It separates each sentence in tokens, using NLTK's word tokenizer. For each token, it determines that it is a drug if it is fully upper-case or if it ends with a common drug suffix (”azole”, ”idine”, ”amine” or ”mycin”). It follows the logic that many drug brands are in upper-case, so in the first case the returned type is brand. Considering its simplicity, it has high F1 score for brand names (0.48; a high percentage of fully capitalized words in the sentences are brand names). The second rule is based on the fact that approximately two thirds of drug names end with the mentioned suffixes, so it

returns that the type of the entity is a base drug, not a brand. It has a very high precision for base drugs (0.94; almost everything recognized as having type drug it is indeed of this type), but poor recall (many entities with the same suffixes are not drugs). This is a very simple model, and since it does not consider at all the group and non-human drug types the F1 score for them is 0, which decreases the average F1 to 0.14 (with precision being 0.42 and recall 0.1).

2.2 Naive Bayes model

The multinomial Naive Bayes is a simple statistical model with a small set of categorical features, used to train a model that attempts to learn whether tokens (words) of sentences are drugs, and their type in such case. The used features, generated for each word in each sentence, are all categorical: POS tag, syntactic chunk, named entity type, whether the word ends in one of the suffixes of the trivial model, a feature describing the orthography of the word (whether it is all- capitalized, starts with upper-case, all-numeric, combines letters and numbers or has another format), whether it contains a hyphen, whether the word is present in the external DrugBank list of drugs, or whether it appears in the NIH list. Features with higher cardinality, such as word or lemma, were discarded due to memory errors (based on the large number of possible values when each feature's categorical values were converted into a binary vector representation resembling a bag of words).

The Naive Bayes model, that uses a 0.01 learning rate, is not able to fully model the sequential relations between words in sentences, but uses the features to build a set of probabilities that naively assumes to be independent of each other. Therefore, it is less accurate than other models that take the sequential nature of sentences into account, such as CRF. In particular, the Naive Bayes model obtains a F1 score of 0.36 (with 0.37 as precision and 0.38 recall). It is particularly good on types drug and brand (with exact matching F1 of 0.71 and 0.7, respectively), but specially bad on the less common group and non-human types, with results close to 0.

2.3 CRF model

Conditional Random Fields are sequential models, so they are very suitable to analyze sentences: to classify each word as drug or not (and its specific type if it is a drug), they can take into account the local context of the word, i.e. the previous and next words. The parameters of the CRF model were optimized using the validation set. The configuration with the highest F1 results was very similar to the default setup of CRF-Suite (library used for the implementation). The minimum frequency of features to be considered was set to 1 (otherwise part of the information was lost), and the regularization coefficient was set to 0.15 (lower values tended to overfitting, with lower training loss but higher validation error). The maximum iteration number is set to 1000, but normally not reached (early stopping takes place after 10 iterations without improvement). The initial learning rate is calculated performing 20 trials, and decreased with a rate of 2 throughout iterations. The sentences were tokenized using the biomedical-specialized Genia tagger, also used to produce POS tags and syntactic chunks and named entities. The performance, nevertheless, was equivalent to that of using NLTK tagging tools.

A comprehensive set of feature types was defined for the CRF model, at word level, all of them corresponding to categorical information, and being described as "$< feature\_type >=< category >$". Internally, CRF generates a binary feature out of each different category, being 1 if explicitly told to appear for one word, 0 otherwise. Each of the feature types is generated for the current word and the immediately previous and next word. A smaller set of feature types (word form and lower-case word form) are also generated for the two-words previous and next words. Most features are based on the ones explained by [1], which surpassed the results of the winners of the original competition by 7.86%, according to the paper.

While constructing the learning system, it was observed that some feature types increased the performance in the validation executions in all of the numerous manual combinations tested (enabling some features and disabling others, at the time of adding a new one). Those clearly beneficial features were added to a White-List, so that they would be always considered in the latter tests that would generate multiple combinations of features to determine the most accurate ones for the validation set. The rest of features had a slightly negative, barely positive or negligible impact in the initial manual tests, so were considered of being potentially convenient or inconvenient de-

pending on their combination with other features. Therefore, they were not added to the White-List, and considered in the different set of combinations in the later feature selection step performed on the validation set.

The complete list of feature types is the following ([W-L] states that the features are in the White-List, but "*" indicates that only for the current word, not for surrounding ones):

- Word form: The word as it is found in the text. [W-L].
- Lower-case form: The word in lower case. [W-L]
- Affixes: different features for the prefixes and suffixes containing the first and last 3, 4 and 5 letters of the word, respectively. [W-L for prefix of 3 letters, W-L* for rest, except affixes of 5, that are discarded]
- Orthographic feature, with different categories (the word is fully capitalized, starts with upper-case, it is all-numeric, combines letters and numbers, or has another format). [W-L]
- Contains hyphen binary feature (present or not). [W-L*]
- Binary features indicating whether the word is present in an external list of drug names. The databases have been obtained from three sources: DrugBank (which comes from one of the same sources as the DDI database, but does not contain directly the test information, so it was not considered inappropriate, and it fact did not have as much impact as other databases), US National Institute of Health (NIH) and US Food Drug Administration (FDA). [W-L*, except FDA]
- Part-Of-Speech tag of the word. [W-L*]
- Lemma: base word from which the word was produced. [W-L*]
- Syntactic chunk of the word, stating which kind of syntactic construction they are part of (a word can be in the beginning, inside or outside one, i.e. using the BIO model). [W-L*]
- Named entity tag for the word (a word can be in the beginning, inside or outside one, i.e. using the BIO model). [W-L*]
- Word shape feature, describing in a generalized way the structure of the word characters, replacing with X the upper-case letters, using x for lower- case ones, 0 for digits and an O for other characters. [W-L*]
- Brief word shape feature, same as previous but removing consecutive repetitions of the same character types. [W-L*]
- Word embedding cluster identifier. Word embeddings based on Word2Vec were used. In particular, a pre-trained set of embeddings (trained in 27 milion Pubmed/Medline articles, containing over 2 milion words) produced by Athens University of Economics and Business. It represents each word as a floating-point vector of 200 dimensions. After loading the model, each word is clustered using K-Means (in a batched way to avoid memory errors due to the high space complexity). This way, semantic classes of similar words are obtained. Different experiments were run and tested on the validation set, and it was found that a randomly-initialized K-Means setup with 450 classes produced the best results, curiously yielding better F1 than K-Means++, with a higher time complexity and usually more accurate results according to the clustering literature. [W-L*]
- Beginning of sentence binary feature, stating if this word is the first of the sentence. [W-L*]
- End of sentence binary feature, stating if this word is the last of the sentence. [W-L*]

To predict the tags of the words, whether they are drugs or not (and their type if they are drugs), two different schemes were tested: BIO and BILOU. B stands for beginning term (of a drug in this case), I is inside, L is last, U is unit-term drug entity and O is outside (not a drug). If BIO is used, the U case is covered by B, and L is covered by I. Similar results were obtained with both models, with BIO being slightly more accurate (F1 0.74 instead of BILOU's 0.7), which is surprising at first since BILOU can learn more details of the sequentiality, but for this data set is has proven to be more suitable to learn the more generalized information provided by BIO.

The selection of the best features was tested by generating a considerable number of combinations

Table 1: Test results of the best CRF model for Task 9.1.

Field	Precision	Recall	F1
Strict matching	0.82	0.72	0.77
Exact matching	0.90	0.79	0.84
Partial matching	0.90	0.81	0.86
Type matching	0.86	0.75	0.80
Exact matching on drug	0.92	0.84	0.88
Exact matching on brand	1.00	0.90	0.95
Exact matching on group	0.88	0.84	0.86
Exact matching on non-human	1.00	0.15	0.26
Macro-average	0.95	0.68	0.74

(50) to construct the model with part of the training set and testing in the remaining part, validation set. The whole training set was then used to build final models that were tested with the test set, to obtain the definitive F1 scores. All the combinations contain the White-List features, and subsets of the other features. Testing the totality of possible combinations of the non-White-List features would have not been computationally feasible (the complexity is exponential), but the obtained results are very satisfactory.

At the end of this document, Table 3 depicts the combinations that yielded the highest validation F1 results (0.74), and their corresponding validation results (ranging from 0.72 to 0.74). It can be observed that just using the White-List features allows to obtain 0.74 in the test set with precision 0.95 and recall 0.68, as shown in Table 1. Using Occam's razor principle of simplicity, this is considered to be the best model, with meaningless differences in precision and recall with other feature sets that also achieve 0.74 F1. This model obtains very high F1 results in all strict and partial matching cases (between 0.8 and 0.95), except for non-drug type (0.21), which is less common than other types and more difficult to model. 0.74 F1 is significantly higher than winner WBI team's 0.643 in the original competition.

3 Task 9.2: Extraction and Classification of Drug-Drug Interactions

3.1 Baseline rule-based model

This trivial model is an extended version of the third laboratory session material of the AHLT course. For each pair of drugs in a sentence annotated as the first and second drug whose interaction should be predicted, all the tokens between the two drugs are found, after tokenizing with NLTK. Each of these words is checked, to see if it is present in a dictionary of key-words, that maps words to their interaction type, based on the examples given in the official "Annotation Guidelines". For instance, "should" is associated with the advise type, "elevation" with effect, "inhibit" with mechanism and "interaction" with the default interaction type. If none of the words is found in the dictionary, then it is considered that the two drugs do not interact. This simple baseline model obtains a precision of 0.25, recall of 0.31 and 0.28 F1-score, which is quite low, but nonetheless higher than the lowest run of a participant in the original competition (0.21 F1). Different variations, such as checking not only the words between the two drugs but also terms before and after, gave worse F1.

3.2 Ensemble of Deep Neural Networks model

The machine learning approach used for DDI classification is an ensemble of multiple Deep Neural Network models, each of them trained with 90% of the training sentences and validated with the remaining 10%. Afterwards, each model is used to predict the class of DDI (no interaction, generic interaction, advise, effect or mechanism), and a majority vote determines the final prediction of the ensemble, i.e. the mode class for each pair of drugs per sentence. The basic principle of ensemble learning is that all the individual classifiers should be capable of obtaining better-than-random accuracy, so that the errors (i.e. due to overfitting) of one model in a certain aspect of the classification are mitigated by the results of the rest of the models.

Drug anonymization is performed as a pre-

processing step, based on previous works [2]. It consists in replacing drug names with generalized strings: the two drugs whose DDI relation should be considered are replaced with "drug1" and "drug2", respectively, and any other drugs in the sentence with "drugother". The motivation of this technique is to focus on the possible interaction between the drugs, not in the particular drugs themselves, so that the model can learn to infer that same or similar texts describing an interaction of two different pairs of drugs correspond to the same DDI class (or no interaction). The sentence words are found with the NLTK word tokenizer, and the ones containing non-alphanumeric characters are discarded (with the exception of hyphens, that are accepted, e.g. "false-positive"). All tokens are converted to lower-case for case independence. Several cases of annotation errors were found in the original data, where the drug names appearing in the sentences were not fully written in the drug entity annotations, especially drugs appearing in plural but showing the singular form in the drug entity, or not including prefixes that appeared in the text. These errors were reflected by generating strings such as "drugothers" (when a drug appeared in plural in the text but not in the entity annotation), that were discarded to prevent unexpected behavior.

Since this is a multi-class classification problem, categorical cross-entropy is used as the loss function. Many optimizing methods have been tested (SGD, Adam, Adadelta, etc.), but none of them as accurate as Nadam, thanks to its use of Nesterov momentum. The different models have different architectures, but share the fact that they start with a word embedding layer, and end with a 5-unit fully-connected layer with softmax activation function (each unit corresponds to one of the possibles DDI classes, including no interaction): the class with the highest score is selected as predicted. The embedding layer learns a real-valued vector for each unique word in the DDI corpus. The embeddings are not randomly initialized; instead, their initial value is provided by the 200-dimensional word vectors of the pre-trained Publine/Medline embeddings for each word in the DDI vocabulary. The word vectors are fine-tuned by each model.

The hidden layers of ensemble models include Convolution layers, since they can detect patterns in the data (in this case, relationships between words in sentences that appear close to each other, in a single dimension), and Recurrent layers, since they use the sequential information of data series (words in sentences in this case) for the prediction task. Some models have Convolution layers but not Recurrent layers, while others have only Recurrent ones, and other models have both types of layers. Different models have different number of convolution filters (50, 100, 150, 300, 500), mask size (2, 3, 4, 8, 12, 20) and different recurrent architecture (one of two common state-of-the-art approaches, LSTM or GRU), and number of recurrent units (32, 64, 128, 256). Max-pooling layers are used after Convolution layers to reduce the dimensionality of the data, and select the stronger, more distinctive filter matches. Most models use additional fully-connected (dense) layers with rectified-linear activation (to avoid exploding or vanishing gradients), with a variable number of units (250, 500, 1000), commonly followed by Dropout layers to discard part of the neurons in each batch and increase generalization and redundancy, to mitigate overfitting; different dropout ratios are found in the models (0, 0.1, 0.2, 0.3, 0.5). Occasionally, Flatten layers are required to make consecutive layers compatible, by adapting the number of dimensions.

One of the models of the ensemble achieves 0.60 F1 individually, which is higher than any other model alone. It is a CNN represented in Figure 1, whose architecture starts with the previously described embedding layer, followed by a Dropout layer with dropout ratio of 0.3, a 1-D Convolution layer with 100 filters and mask size 4, followed by Max Pooling layer, and afterwards a 500-unit fully-connected Dense layer, another 0.3 Dropout layer and the final Dense layer with softmax activation function to decide between the 4 classes of DDI or no DDI.

The combination of 8 ensembles (including the model mentioned in the previous paragraph and other CNNs, RNNs and hybrid models, as previously described) produces a macro-average F1 of 0.63, while the F1 for determining if two drugs interact in a sentence is 0.69, and the F1 for the classification of DDI cases is 0.63. This information, as well as the precision, recall and F1 for each DDI type, is shown in Table 2. These results are only outperformed by one team in the original competition, FBK-irst, that obtained a macro-average F1 of roughly 0.65 in their best run, using SVMs.

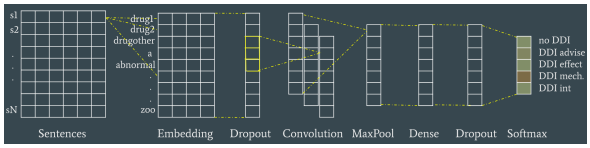

Figure 1: Architecture of the submodel with highest F1.

Table 2: Test results of the best model for Task 9.2.

Field	P	R	F1
Partial Evaluation: only detection of DDI	0.74	0.67	0.69
Detection and Classification of DDI	0.67	0.59	0.63
DDI with type mechanism	0.69	0.59	0.64
DDI with type effect	0.63	0.61	0.62
DDI with type advise	0.69	0.68	0.68
DDI with type int	0.97	0.30	0.46
Macro-average	0.74	0.55	0.63

4 Conclusions and Future work

The machine learning solutions developed for the tasks, namely a CRF model (Task 9.1) and an ensemble of Deep Neural Network classifiers (Task 9.2), have significantly outperformed the baselines. Therefore, it can be concluded that advanced approaches that extract features from the data and generalize complex models to learn a given task (drug name recognition and classification or DDI detection and classification) provide results of much higher quality than simple sets of rules.

The CRF model has significantly outperformed the results of the original participants of Task 9.1 (an F1 of 0.74, compared to the winner's 0.64). Nevertheless, some authors propose additional mechanisms to slightly improve the results. In this work a limited set of feature sets among the exponential total number of combinations was used, while other authors [1] use sub-optimal feature selection strategies, e.g. based on Information Gain or Chi-squared. The ensemble of Neural Networks used in Task 9.2 can be further improved by incorporating attention mechanisms to the CNNs to focus on individual words at each pass (and their relationship with others), or using genetic algorithms to evolve different populations of ensembles (with validation F1 as the fitness function) to select the most suitable combination.

References

[1] Shengyu Liu, Buzhou Tang, Qingcai Chen, Xiaolong Wang, and Xiaoming Fan. Feature engineering for drug name recognition in biomedical texts: Feature conjunction and feature selection. *Computational and mathematical methods in medicine*, 2015, 2015.

[2] Shengyu Liu, Buzhou Tang, Qingcai Chen, and Xiaolong Wang. Drug-drug interaction extraction via convolutional neural networks. *Computational and mathematical methods in medicine*, 2016, 2016.

[3] Isabel Segura Bedmar, Paloma Martínez, and María Herrero Zazo. Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (ddiextraction 2013). Association for Computational Linguistics, 2013.

[4] María Herrero-Zazo, Isabel Segura-Bedmar, Paloma Martínez, and Thierry Declerck. The ddi corpus: An annotated corpus with pharmacological substances and drug–drug interactions. *Journal of biomedical informatics*, 46(5):914–920, 2013.

Table 3: Features that yield the best validation F1-score (0.74) in Task 9.1 with their results when evaluated on the test set.

Features	**P valid**	**R valid**	**F1 valid**	**P test**	**R test**	**F1 test**
W-L+form_prev+form_lower_prev+suf3_prev+suf4_prev+orthography_prev+contains_hyphen_prev +suf3_next+suf4_next+orthography_next+contains_hyphen_next+is_in_drug_set_nih_prev+chunk_prev+embedding_class_prev+is_in_drug_set_fda_next+is_in_drug_set_drug_bank_next+chunk_next+named_entity_next+word_shape_next+brief_word_shape_next+embedding_class_next	0.93	0.69	0.74	0.94	0.67	0.72
W-L	0.93	0.68	0.74	0.95	0.68	0.74
W-L+form_prev+form_lower_prev+suf3_prev+suf4_prev+orthography_prev+contains_hyphen_prev +suf3_next+suf4_next+orthography_next+contains_hyphen_next+is_in_drug_set_fda+named_entity_pre +embedding_class_prev+is_in_drug_set_drug_bank_next+named_entity_next+word_shape_next+brief_word_shape_next	0.93	0.68	0.74	0.95	0.68	0.74
W-L+suf3_prev+suf4_prev+orthography_prev+contains_hyphen_prev+suf3_next+suf4_next+orthography_next +contains_hyphen_next+is_in_drug_set_fda+chunk_prev+is_in_drug_set_drug_bank_next+chunk_next+word_shape_next+brief_word_shape_next	0.93	0.68	0.74	0.94	0.68	0.73
W-L+suf3_prev+suf4_prev+orthography_prev+contains_hyphen_prev+suf3_next+suf4_next+orthography_next+contains_hyphen_next+is_in_drug_set_fda+chunk_prev+is_in_drug_set_fda_next+is_in_drug_set_drug_bank_next+chunk_next +named_entity_next+embedding_class_next	0.93	0.68	0.74	0.95	0.69	0.73
W-L+suf3_next+suf4_next+orthography_next+contains_hyphen_next+is_in_drug_set_fda+is_in_drug_set_nih_prev+is_in_drug_set_fda_prev+word_shape_prev+brief_word_shape_prev+is_in_drug_set_nih_next+is_in_drug_set_fda_next+is_in_drug_set_drug_bank_next+chunk_next+brief_word_shape_next	0.93	0.68	0.74	0.94	0.69	0.74
W-L+suf3_prev+suf4_prev+orthography_prev+contains_hyphen_prev+suf3_next+suf4_next+orthography_next+contains_hyphen_next+is_in_drug_set_fda+is_in_drug_set_drug_bank_prev+named_entity_prev+brief_word_shapeprev+embeddingclass_prev+is_in_drug_set_fda_next+is_in_drug_set_drug_bank_next+chunk_next+named_entity_next+brief_word_shape_next	0.93	0.68	0.74	0.95	0.68	0.74

Features	**P-valid**	**R-valid**	**F1-valid**	**P-test**	**R-test**	**F1-test**
W-L+suf3_prev+suf4_prev+orthography_prev+contains_hyphen_prev+suf3_next+suf4_next+orthography_next+contains_hyphen_next+is_in_drug_set_fda+is_in_drug_set_fda_prev+is_in_drug_set_drug_bank_prev+is_in_drug_set_fda_next+is_in_drug_set_drug_bank_next+chunk_next+named_entity_next+word_shape_next+brief_word_shape_next	0.93	0.68	0.74	0.95	0.68	0.73
W-L+suf3_prev+suf4_prev+orthography_prev+contains_hyphen_prev+suf3_next+suf4_next+orthography_next+contains_hyphen_next+is_in_drug_set_fda+is_in_drug_set_fda_prev+chunk_prev+word_shape_prev+embedding_class_prev+is_in_drug_set_nih_next+is_in_drug_set_drug_bank_next+chunk_next+brief_word_shape_next+embedding_class_next	0.93	0.68	0.74	0.93	0.68	0.73
W-L+suf3_prev+suf4_prev+orthography_prev+contains_hyphen_prev+suf3_next+suf4_next+orthography_next+contains_hyphen_next+is_in_drug_set_fda+is_in_drug_set_nih_prev+chunk_prev+brief_word_shape_prev+embedding_class_prev+is_in_drug_set_nih_next+brief_word_shape_next	0.93	0.68	0.74	0.94	0.69	0.74
W-L+suf3_prev+suf4_prev+orthography_prev+contains_hyphen_prev+suf3_next+suf4_next+orthography_next+contains_hyphen_next+is_in_drug_set_fda+is_in_drug_set_nih_prev+is_in_drug_set_fda_prev+is_in_drug_set_drug_bank_prev+brief_word_shape_prev+is_in_drug_set_fda_next+chunk_next+brief_word_shape_next	0.93	0.68	0.74	0.95	0.68	0.74
W-L+suf3_prev+suf4_prev+orthography_prev+contains_hyphen_prev+suf3_next+suf4_next+orthography_next+contains_hyphen_next+is_in_drug_set_fda+is_in_drug_set_nih_prev+is_in_drug_set_fda_prev+is_in_drug_set_drug_bank_prev+chunk_prev+named_entity_prev+word_shape_prev+brief_word_shape_prev+embedding_class_prev+is_in_drug_set_fda_next+is_in_drug_set_drug_bank_next+word_shape_next+brief_word_shape_next	0.92	0.68	0.74	0.93	0.68	0.73