# Boosting drug named entity recognition using an aggregate classifier

Ioannis Korkontzelos*[a], Dimitrios Piliouras†[a], Andrew W. Dowsey‡[b,c], and Sophia Ananiadou§[a]

[a]National Centre for Text Mining (NaCTeM), School of Computer Science, The University of Manchester, Manchester Institute of Biotechnology, 131 Princess Street, Manchester M1 7DN, United Kingdom

[b] Centre for Endocrinology and Diabetes, Institute of Human Development, Faculty of Medical and Human Sciences, The University of Manchester, Manchester, United Kingdom

[c]Centre for Advanced Discovery and Experimental Therapeutics (CADET), University of Manchester and Central Manchester University Hospitals NHS Foundation Trust, Manchester Academic Health Sciences Centre, Oxford Road, Manchester M13 9WL, United Kingdom

**Abstract**

**Objective**: Drug named entity recognition (NER) is a critical step for complex biomedical NLP tasks such as the extraction of pharmacogenomic, pharmacodynamic and pharmacokinetic parameters. Large quantities of high quality training data are almost always a prerequisite for employing supervised machine-learning techniques to achieve

*email: Ioannis.Korkontzelos@manchester.ac.uk, *corresponding author*
telephone: +44 (0)161 306 3094
†email: piliourd@cs.man.ac.uk
‡email: Andrew.Dowsey@manchester.ac.uk
§email: Sophia.Ananiadou@manchester.ac.uk

high classification performance. However, the human labour needed to produce and maintain such resources is a significant limitation. In this study, we improve the performance of drug NER without relying exclusively on manual annotations.

**Methods**: We perform drug NER using either a small gold-standard corpus (120 abstracts) or no corpus at all. In our approach, we develop a *voting system* to combine a number of heterogeneous models, based on dictionary knowledge, gold-standard corpora and silver annotations, to enhance performance. To improve recall, we employed genetic programming to evolve 11 regular-expression patterns that capture common drug suffixes and used them as an extra means for recognition.

**Materials**: Our approach uses a dictionary of drug names, i.e. DrugBank, a small manually annotated corpus, i.e. the pharmacokinetic corpus, and a part of the UKPMC database, as raw biomedical text. Gold-standard and silver annotated data are used to train maximum entropy and multinomial logistic regression classifiers.

**Results**: Aggregating drug NER methods, based on gold-standard annotations, dictionary knowledge and patterns, improved the performance on models trained on gold-standard annotations, only, achieving a maximum F-Score of 95%. In addition, combining models trained on silver annotations, dictionary knowledge and patterns are shown to achieve comparable performance to models trained exclusively on gold-standard data. The main reason appears to be the morphological similarities shared among drug names.

**Conclusion**: We conclude that gold-standard data are not a hard requirement for drug NER. Combining heterogeneous models build on dictionary knowledge can achieve similar or comparable classification performance with that of the best performing model trained on gold-standard annotations.

**Keywords:** named entity annotation sparsity, gold-standard vs. silver-standard annotations, named entity recogniser aggregation, genetic-programming-evolved string-similarity patterns, drug named entity recognition

# 1. Introduction

Named entity recognition (NER) is the task of identifying members of various semantic classes, such as *persons*, *mountains* and *vehicles* in raw text. In biomedicine, NER is concerned with classes such as *proteins*, *genes*, *diseases*, *drugs*, *organs*, *DNA sequences*, *RNA sequences* and possibly others [1]. Drugs (as pharmaceutical products) are special types of chemical substances highly relevant for biomedical research. A simplistic and naive approach to NER is to directly match textual expressions found in a relevant lexical repository against raw text. Even though this technique can sometimes work well, often it suffers from certain limitations. Firstly, its accuracy heavily depends on the completeness of the dictionary. However, as terminology is constantly evolving, especially in bio-related disciplines, producing a complete lexical repository is not feasible. Secondly, direct string matching overlooks term ambiguity and variability [2]. On one hand, ambiguous dictionary entries refer to multiple semantic types (term ambiguity), and therefore contextual information needs to be considered for disambiguation. On the other hand, several slightly different tokens may refer to the same semantic type (term variability). Typically, to address these issues, statistical learning models are deployed for NER.

In such approaches, NER is formalised as a classification task in which an input expression is either classified as an entity or not. Supervised learning methods are reported to achieve superior performance than unsupervised ones, but previously annotated data are essential for training supervised models [2]. Data annotated by human curators are of high quality and guarantee best results in exchange for the cost of manual effort. For these reasons, they are also known as gold-standard data. Due to the cost of manual annotations, corpora for NER are often of limited size and for particular domains.

Drugs are referred to by their chemical name, generic name or brand name. Since the chemical name is typically complex and a brand name may not exclusively identify a drug once the relevant patents expire, a unique non-proprietary name for the active ingredient is devised for standardised

scientific reporting and labelling. This generic name is negotiated when the drug is approved for use, as the nomenclature is tightly regulated by the World Health Organization (WHO) and local agencies such as the U.S. Food and Drug Administration (FDA) and the European Medicines Agency. Several criteria are assessed, such as ensuring the drug action fits the naming scheme, ease of pronunciation and translation, and differentiation from other drug names to avoid transcription and reproduction errors during prescription [3]. Since the naming scheme, assessment criteria and cross-border synchrony have developed organically over the years, there is neither a definitive dictionary nor syntax of drug names.

In this study, we investigate methods for achieving high performance in drug name recognition in cases where either very limited or no gold-standard training data is available. Our proposed method employs a *voting system* able to combine predictions from a number of diverse recognisers. Moreover, genetic programming is used to evolve string-similarity patterns based on common suffixes of single-token drug names occurring in the DrugBank database [4]. Subsequently, these patterns are used to compile regular expressions in order to generalise dictionary entries in an effort to increase coverage and tagging accuracy.

We compare the performance of our method with several state-of-the-art NER approaches in recognising manually annotated drug names in the PK corpus [5]. Where no gold-standard data is available, the proposed method is shown to achieve competitive performance. In particular, the performance achieved without gold-standard data is comparable with the performance of the model aware of gold-standard annotations.

The rest of this paper is organised as follows: section 2. summarises previous work on drug NER and methods for dealing with data sparsity in general NER. Section 3. describes the dictionaries and data used in our experiments, as well as the experimental methodology followed. Sections 4. and 5. present and discuss the experiments and their results. Finally, section 6. concludes the paper.

## 2. Related work

NER is a large, well-studied field of natural language processing (NLP) [6]. Most publications address it as a supervised task, i.e. the procedure of training a model on annotated data and then applying it to new text. In the past, several evaluation challenges have taken place on recognising entities of the general domain [7–10] as well as scientific domains [2, 11, 12]. In contrast, research related with Drug NER is limited [13–15]. Very recently, an evaluation challenge that focussed exclusively on drug name recognition and drug-drug interactions has taken place [16].

As a result of the collaborative annotation of a large biomedical corpus project [17], a large-scale biomedical silver standard corpus has been produced. It contains annotations resulting from the harmonisation of named entities (NEs) automatically recognised by five different tools, namely, Whatizit [18], Peregrine [19], GeNO [20], MetaMap [21] and I2E [22]. Apart from names of chemicals and drugs, proteins, genes, diseases and species names were also tagged by these tools in the 174,999 MEDLINE abstracts comprising the corpus. Approximately half a million NE annotations for each semantic category are contained in the resulting harmonised corpus which is publicly available. It has been used for the 2 annotation challenges [23].

Dictionaries and ontologies have been used extensively as the basis to generate patterns and rules for NER. Tsuruoka *et al.* [24] used logistic regression to learn a string similarity measure from a dictionary, useful for soft-string matching. Kolarik *et al.* [25] used lexico-syntactic patterns to extract terms. Patterns are similar to the ones introduced in [26] and contain drug names and directly related drug annotation terms found in DrugBank. Then, these patterns were applied to MEDLINE abstracts, to add annotations of pharmacological effects of drugs. Similar methods have also been applied for recognising drug-disease interactions [27] and interactions between compounds and drug-metabolising enzymes [28]. Hettne *et al.* [29] developed a rule-based method intended for term filtering and disambiguation. They identify names of drugs and small molecules by incorporating several dictionaries such as the UMLS (`nlm.nih.gov/research/umls`, accessed: 15

April 2015), MeSH (nlm.nih.gov/mesh, accessed: 15 April 2015), ChEBI (www.ebi.ac.uk/chebi, accessed: 15 April 2015), DrugBank (drugbank.ca, accessed: 15 April 2015), KEGG (www.genome.jp/kegg, accessed: 15 April 2015), HMDB (hmdb.ca, accessed: 15 April 2015) and ChemIDplus (chem.sis.nlm.nih.gov/chemidplus, accessed: 15 April 2015). An earlier system, EDGAR [30], extracts genes, drugs and relationships between them using existing ontologies and standard NLP tools such as part-of-speech taggers and syntactic parsers.

A popular means of dealing with data sparsity in NER is to generate data semi-automatically or fully automatically. Although, the resulting data is of lower quality than gold-standard annotations, supervised learners can benefit largely from large volumes of data, since they are based on annotation statistics. Towards the same ultimate goal, our approach aims to overcome the restrictions of data sparsity or unavailability in the biomedical domain. Usami *et al.* [31] describe an approach for automatically acquiring large amounts of training data from a lexical database and raw text that relies on reference information and coordination analysis. Similarly, noisy training data was obtained by using a few manually annotated abstracts from FlyBase (flybase.org, accessed: 15 April 2015) [32, 33]. The approach uses a bootstrapping method and context-based classifiers to increase the number of NE mentions in the original noisy training data. Even though they report high performance, their method requires some minimum curated seed data. Similarly, Thomas *et al.* [34] demonstrated the potential of distant learning in constructing a fully automated relation extraction process. They produced two distantly labelled corpora for protein-protein & drug-drug interaction extraction, with knowledge found in databases such as IntAct [35] for genes and DrugBank [4] for drugs.

*Active learning* is a framework that can be used for reducing the amount of human effort required to create a training corpus [36, 37]. The most informative samples are chosen from a big pool of human annotations by a maximum likelihood model in an iterative and interactive manner. It has been shown that active learning can often drastically reduce the amount of training data necessary to achieve the same level of performance compared

to pure random sampling [38]. A similar approach, *accelerated annotation* [39], allows to produce NE annotations for a given corpus at reduced cost. In contrast to active learning, it aims to annotate all occurrences of the target NEs, thus minimising the sampling bias. Despite the similarities between the two frameworks, their goals are different. While active learning aims to optimise the performance of the corresponding tagger, accelerated annotation aims to construct an unbiased NE annotated corpus.

# 3. Methods and data

In this section we present our aggregate classifier for recognising drug names and the necessary resources.

## 3.1. Methodology

To classify labels of tokens, we used two classifiers, a *maximum entropy* (MaxEnt) model, also known as *multinomial logistic regression* [40], and a perceptron classifier [41]. MaxEnt classifiers assume that the best model parameters are the ones for which each feature's predicted expectation matches its empirical expectation and classify instances so that the conditional likelihood is maximised. In other words, MaxEnt maximises entropy while conforming to the probability distribution drawn by the training set. Perceptron is a linear classifier that tunes the weights in a network during the training phase, so as to produce the desired output. The perceptron method is guaranteed to locate the combination of weights that solve the problem, if such a combination exists. We used standard implementations of MaxEnt and Perceptron, parts of the Apache openNLP project (`opennlp.apache.org`, accessed: 15 April 2015).

For both classifiers, we used the same feature set, described below. For each token, we consider as features:

- **tokens**: the current and $\pm 2$ tokens
- **character n-grams**: $\pm 2$ tokens

- **sentence**: a binary feature indicating whether the token appears at start or end of a sentence
- **binary token type features** of the current and $\pm 2$ tokens, shown in table 1
- **previous map**: a binary feature indicating whether the current token was previously seen as a NE
- **prefix** and **suffix** of the current token
- **dictionary**: a binary feature indicating whether the current token exists in the dictionary

We attempt to aggregate predictions from dictionaries and NER systems, under the fundamental hypothesis that the combined output might improve over the results of single classifiers deployed as standalone. Our aggregate classifier is compatible with any dictionaries and recognition systems, and could be applied in other domains and sequence recognition tasks.

We developed a simple voting-system assuming that the predictions of dictionaries are more reliable than predictions of machine learners. As a result, the algorithm accepts dictionary predictions as valid if they exist. This assumption is not true, if a dictionary contains non-drug entities but, since dictionaries are produced manually, we consider them ideal. Ambiguous NEs might also affect the validity of this assumption. We observed very little such ambiguities in our dictionary, DrugBank, thus, we accept the hypothesis to hold in the domain of drug NEs.

Algorithm 1 summarises the voting system. In short, it starts with an empty list L and iterates over all sentences and tokens of the input text. For each token, it queries available dictionaries or regular expression patterns and accepts positive answers as valid. Otherwise, it considers sequentially each model's opinion regarding whether the current token is a drug entity or not. Whenever a model positively recognises a drug-name, we store the name along with the confidence of the prediction in a map. After considering the predictions of all models, we store the positive prediction with the highest confidence, if such a prediction exists, and proceed to the next token. Predictions of dictionaries or regular expression patterns are assigned

100% confidence.

After processing all tokens of a sentence, any intersecting or overlapping spans are removed according to the following rules. For predictions that cross into each-other, we discard all but the first one. For nested predictions, only the maximal one is kept. Upon algorithm completion, L should contain a sequence of maps, each representing the predictions on a single sentence.

At a second experimental stage, we de-constructed the dictionary into 2 distinct models: (a) a model trained on text solely annotated by the dictionary, and (b) an evolved set of string-patterns that attempts to accurately cover common suffixes of single-token drug names.

For evaluation, we used the standard information retrieval metrics: precision, recall and F-Score ($\mathbf{F}_1$) [42].

## 3.2. Data

The proposed method requires two types of resources: (a) one or more comprehensive lexical repositories, such as dictionaries or lexicons. (b) large amounts of raw text in the domain of interest, which is drugs for the current study. Supplementally, a small gold-standard corpus may enhance NER performance if available.

Our method could potentially be applied to recognise any type of biomedical NEs, such as genes and proteins. We choose to focus on identifying *drug* names, as this domain has been studied to a much smaller extent. In this section, we present the resources that we made available to the algorithm proposed in section 3.1. for experimentation.

### 3.2.1. DrugBank

As our dictionary, we chose to use DrugBank [4] because it is relatively up-to-date and provides a mapping between drug-names and common synonyms. DrugBank currently contains more than $6,700$ entries including $1,447$ FDA-approved small molecule drugs, 131 FDA-approved biotech (protein/peptide) drugs, 85 nutraceuticals and $5,080$ experimental drugs. We pre-processed the dictionary by normalising all official drug terms and linked

them to their synonyms in a key-value data-structure. Each key (drug name) is unique and maps to a single value (a list of synonyms).

### 3.2.2. The pharmacokinetic corpus

The pharmacokinetic corpus (PK) [5] is manually annotated and consists of 240 MEDLINE abstracts annotated and labelled on the basis of MESH terms relevant to pharmacokinetics such as drug names, enzyme names and pharmacokinetic parameters, e.g. clearance. Half of the corpus is intended for training (invivo/invitro-train) and half for testing (invivo/invitro-test). It is freely available at: `rweb.compbio.iupui.edu/corpus` (Accessed: 15 April 2015). As a pre-processing step, all annotations concerning entities other than drugs were removed, since this study is concerned with detecting drug names only.

### 3.2.3. Raw text

Nowadays, acquiring large amounts of raw text is not a difficult task, even for very specialised domains. Public electronic repositories of open-access articles exist for most scientific domains and usually can be queried via RESTful web services. In biomedicine, for example, UK PubMed Central (UKPMC, `europepmc.org`, accessed: 15 April 2015) is an article database which extends the functionality of the original PubMed Central (`ncbi.nlm.nih.gov/pmc`, accessed: 15 April 2015) repository. For the purposes of this study, we used a small subset of the entire UKPMC database which includes more than two million papers. The sample we used was created by Mihăilă and Navarro [43], totalling 360 pharmacology and cell-biology related articles. As a pre-processing step, the corpus was sentence-split and tokenised. Part-of-speech tagging was omitted from the process since we did not plan to use the part-of-speech tags as features during training.

## 4. Experimental results

The first set of experiments, in section 4.1., considered the entire set of gold-standard annotations. In the second set, we assess the importance of silver data, i.e. data annotated by recognising dictionary entries on raw text. We evaluate the performance of our classifier trained on silver data only, and we also combine gold and silver data to measure whether the combination can boost performance. In succession, we investigate how we can produce string similarity patterns based on dictionary knowledge to further increase recall. Finally, we investigate how the proposed aggregate classifier performs in absence of gold-standard annotations.

### 4.1. Baselines

Firstly, we tested how the dictionary performs as a single recogniser, including or excluding synonyms. Secondly, we trained two NE recognisers, namely a MaxEnt and a perceptron classifier, on half the PK corpus (invivo/invitro-train) and tested them on the other half (invivo/invitro-test). Finally, we used our prediction aggregation algorithm to combine predictions originating from the dictionary, with predictions originating from the classifiers.

Table 2 presents the results from our baseline experiments. It is worth noting that the pure dictionary-based approach is not 100% precise as our voting system assumes. Careful error analysis revealed that there are at least two entities, i.e. "nitric oxide" and "tranylcypromine" that have not been tagged in the gold-standard corpus. Consequently, the evaluator marks them as false-positives while, in fact, they are perfectly correct predictions. Another interesting observation is that including synonyms causes precision to degrade. Synonyms in DrugBank often include acronyms, which have not been tagged appropriately in the test corpus. As before, the evaluator classifies them as false-positives.

In general, we can see that both the dictionary and the classifiers exhibit very high precision and good recall, whereas combining the two has a minimal positive effect on overall performance. The perceptron classifier, despite training significantly faster, consistently showed inferior performance

in comparison with MaxEnt.

Unfortunately, no other experimental results on the exact data that we experimented with have been published. However, to put our results into perspective, we can consider the results of a recent evaluation challenge, SemEval-2013 Task 9: DDIExtraction [16]. Its first subtask was concerned with recognising and classifying drug names. Several participating systems aimed at recognising generic and branded drug names, among other entities. Table 3 show the exact-matching precision, recall and F-Score achieved by the best performing systems in terms of F-Score in the categories of generic and branded drug names. The DDIExtraction 2013 task was evaluated on the DDI corpus, which consists of 784 DrugBank texts and 233 MEDLINE abstracts and was manually annotated [44]. Although the data used in this work are not identical to the DDI corpus, the results in table 3 can be used as indirect baselines. It can be observed that the our baseline results in table 2 are comparable if not slightly better than the best performing systems the DDIExtraction2013 task, despite the fact that we trained on significantly smaller and possibly lower-quality data.

Our baseline experiments show that, despite acquiring state-of-the-art precision, there is still space for improvement with regards to recall. High precision indicates that the model extracts some very informative features while training, whereas not so high recall essentially reflects lack of enough training data. Ideally, we would need more gold-standard annotations, however, as discussed previously, this is not always feasible.

### 4.2. Combining heterogeneous models

Attempting to improve recall, we trained separate models purely on silver data, i.e. data annotated by direct string-matching dictionary entries. Annotation coverage ultimately depends on the quality of the dictionary, its coverage and how up-to-date it is. DrugBank is a good candidate for this task, as it is a comprehensive dictionary of drugs and also freely available. The 360 full papers mentioned in section 3.2.3. were annotated and partitioned into 30 collections, each one containing 12 items. This was done in an

effort to incrementally check whether the addition of silver annotations has any positive or negative effects on classifier's performance. We found that we had to include all 30 partitions in order to witness some improvement.

Table 6 summarises the results of our experiments. The MaxEnt classifier trained on silver annotation data achieves marginally higher precision and significantly lower recall than the same classifier trained on gold-standard data. This is expected, since the silver annotations reflect the contents of the dictionary, only. Trained on a mixture of gold and silver data, the MaxEnt classifier achieves 0.5% lower precision and 0.3% higher recall than its equivalent trained on gold-standard data.

Including the dictionary boosts the recall of the MaxEnt classifier trained on a mixture of gold-standard and silver annotation data by 1.3% in comparison with its baseline equivalent. The last 2 rows of Table 4 show that all statistics were slightly boosted just by utilising these extra, easy to produce silver annotations.

In all our experiments so far, the best achieved recall is 89.7%, far less than precision, thus, we focus on improving it. Careful examination of false-negatives reveals that most of them are either acronyms (e.g. HMR1766), long chemical descriptions (e.g. 5beta-cholestane-3alpha, 7alpha, 12alpha-triol) or terms whose lexical morphology is particularly different than the usual morphology of drugs (e.g. grapefruit juice). We attempted to capture acronyms by employing a state-of-the-art acronym disambiguator, AcroMine [45], however it did not disambiguate any of the acronyms in question, listed below:

- ANF
- PO4
- HMR1766
- MDZ 1'-OH
- E3174
- RPR 106541
- MDZ 4-OH

Under data sparsity, it is crucial to extract maximum utility from our training set, which necessitates incorporation of features with low occurrence frequency. The MaxEnt and perceptron models we employ do not consider the uncertainty introduced by low frequency training data, hence a frequency threshold value is introduced to control the compromise between precision

and recall. For our experiments we set this threshold at 5, as lower values detrimentally affected precision. As discussed, a number of false-negatives were missed due to their morphology which is different than the usual morphology of drugs. These two facts, suggest that probably some informative features did not qualify due to the frequency threshold value. It should be noted that, in contrast to the MaxEnt classifier, which is probabilistic, the perceptron classifier is not affected by the frequency threshold. The perceptron is essentially a neural network, thus it does not gather probabilities and therefore performs best when no frequency threshold is applied.

Beyond the scope of this paper, there are more sophisticated methods to select important features or tune feature weights to address data sparsity. In general, two smoothing approaches [46] are applied: linear interpolation [47] and back-off models [48–51]. Data sparsity can also be addressed by feature relaxation, based on hierarchical features [52]. A more sophisticated approach to feature weighting would be to employ *Dirichlet regression* [53–55] rather than MaxEnt. Dirichlet regression considers frequencies directly as the dependent variable, rather than probabilities as in multinomial logistic regression. The sum of frequencies for a particular feature represents its "precision". It should be noted that for high frequency data Dirichlet and multinomial logistic regression models behave similarly.

### 4.3. Evolving string-similarity patterns

In this section we aim to improve recall by learning string similarity patterns based on dictionary knowledge. Exploring ways to restore the predictive power the model could have if more training data were available, we develop a mechanism to deal with these easy, yet elusive false-negative cases discussed in the previous section. We attempt to genetically evolve string patterns that can then be used as regular expressions to capture drug names that are not present in the dictionary. We followed a three-step process described below: *genetic programming*, *filtering* and *pattern augmentation*.

Following the work of Tsuruoka *et al.* [24], we also use a form of regression in order to learn common string patterns of drug names. More

specifically, we used genetic programming, also known as symbolic regression, a technique which allows the evolution of programs at the symbolic level [56]. **Genetic programming** is used in this task as a global optimisation algorithm.

The pseudo-random sampling inherent in genetic programming means that no hard guarantees about the final outcome can be made. However, the randomness also enables good coverage of the fitness landscape and therefore avoids falling into local optima, which is essential to solve our problem. Furthermore, the self-driven nature of evolution is robust as it makes little to no assumptions about the fitness landscape, thus mitigating bias during the learning stage, which enables it to produce meaningful solutions where other global optimisation algorithms can falter [56]. Learning by means of evolution is a good fit for our use-case as it allows finding decent solutions with minimal prior knowledge.

Genetic programs assemble variable length program structures from the basic units, i.e. functions and terminals. The assembly occurs at the beginning of a run, when the populations is initialised. In succession, programs are transformed using genetic operators, such as *crossover*, *mutation* and *reproduction*. The algorithm evolves a population of programs by determining which individuals to improve based on their fitness, which is in turn assessed by the fitness-function.

In our implementation, genetic programs were represented as trees that were traversed in a depth-first manner. A *fitness function*, a *function set* and a *terminal set* are required for developing a genetic algorithm. Terminals provide the values that can be assigned to the tree leaves, while functions perform operations on either terminals or on the output of other functions. Typically, the function set of a genetic algorithm that deals with numerical calculations contains the four basic arithmetic operations (+ - * /), while the terminal set contains one-digit non-negative integers, [0-9]. In the current case, the function and terminal sets have to deal with strings. The terminal set contains all latin lowercase letters, [a-z], plus several other characters needed for building meaningful regular expressions, i.e. | * + ? ( ) [ ]. The function set contains several string-manipulating functions, e.g. *split*, *join*

and *concatenate*.

We employed two genetic operators, *crossover* and *mutation*. As illustrated in figure 2, tree-based crossover generates new individuals by swapping subtrees of existing individuals. The population percentage applicable to crossover was set to 35%. Mutation typically operates on one individual. As shown in figure 3, a point in the tree under mutation is chosen and the corresponding subtree is replaced with a new randomly generated subtree. This new subtree is created in the same way, and is subject to the same limitations as the original trees in the initial population. As a matter of future work, more genetic operators can be employed. Although we attempt to employ the simplest genetic operators possible, evolving similarity patterns by genetic programming is the most demanding part of this work, in terms of computational complexity.

Each "organism" in the genetic population is a small program. When executed, the program produces a string that is assigned a score according to the fitness function. For this purpose, all the single-word terms were extracted from DrugBank and were used as "test-data" within the fitness function, which returns the proportion of matches as a measure of fitness. In case the string produced is not a valid regular expression, the candidate receives negative score and will most likely be disregarded in the next generation. For instance, a candidate that matches 50/6,700 terms in DrugBank is obviously fitter than one that matches only 10/6,700 terms, which in turn, is fitter than one whose string does not compile as a regular expression. However, genetic programming did not achieve anything less than 100% error when attempting to match entire tokens, and so we limited the testing scope to the last 4, 5 or 6 characters of each token. This decision was made after observing that word-endings tend to be more similar than word-beginnings in drug names, mainly for conformance with the WHO's international non-proprietary name stem grouping (`who.int/medicines/services/inn/stembook/en`, accessed: 15 April 2015). This had a major positive effect on the population in most executions.

After 200 experiments with 80 generations per experiment and $10,000$ individuals per generation, the 30 best-evolved individuals were selected. Each

individual is a function that builds a string that represents a potentially common suffix, in the form of a regular expression pattern. The pattern produced by the best individual matched 7.3% of the terms in the test-set (130 terms). It should be noted that the evolutionary process evaluates candidates using a list of singletons and not actual sentences. As a consequence, these patterns will most likely introduce false-positives if applied directly on real text, thus, decreasing precision.

In succession, we aim to keep the top performing patterns only, i.e. the least likely to introduce false-positives. This **filtering** can be done either manually or algorithmically. Since the number of patterns is small, the cost of manual checking by a domain expert is limited. Non-experts could also accomplish this task. Instead, we chose to increase the applicability of our approach, we selected the best patterns automatically. We calculated all possible combinations of sets of 5 string patterns and performed an extensive evaluation process where each combination was evaluated only for false-positives on 10 randomly selected paragraphs from the original training set (PK corpus). We selected 5 sets of patterns (25 patterns) which introduced the least false-positives. These 25 patterns were reduced to 11 after removing duplicates and those that would clearly introduce false-positives. For example, the pattern *"m?ine"* was removed because it would recognise *"fluvoxamine"* correctly, but it would also incorrectly recognise as drugs words such as *"examine"*, *"define"*, *"jasmine"* and *"cosine"*. Table 5 shows the 11 best performing patterns, accompanied with the number of matches and an example for each one, while figure 1 shows the tree that corresponds to the best performing pattern.

Finally, in the **pattern augmentation** step, we augmented these 11 patterns by wrapping them as follows:

\b (\d?\,?\d'?\-?)?\w+ <*pattern*>+ \b

The string *"\b"* at the start and end of the pattern, make it applicable to whole words only. The string *"(\d?\,?\d'?\-?)?\w+"* specifies optional triggers, i.e. digits, commas and dashes, mainly for matching hydroxylated compounds. For example if a pattern applies to "midazolam",

it also matches "4-hydroxymidazolam", "4,5-hydroxymidazolam" and "4,5'-hydroxymidazolam". It is common knowledge in biochemistry that all organic compounds go through *oxidative degradation* when they come in contact with air. Hydroxylation is the first step in that process and converts lipophilic compounds into water-soluble (hydrophilic) products that are more readily excreted. We observe many mentions of such compounds in pharmacology papers, and therefore we attempt to capture them with this simple regular expression. The pattern augmentation rule, was chosen manually, introducing a minimal human interaction in this last step.

The genetic programming paradigm parallels nature in that it is a never-ending process. In practise however, and particularly when evolving code, arbitrary complexity is rarely desired because it is very easy for the model to over-fit or start deviating substantially from a good solution approximation. We adopt two simple and widely used termination criteria to address this. We stopped the evolution process (a) after a number of iterations (generations) and (b) by setting a maximum allowed tree depth (10). The patterns were evolved assuming that each will span a single word term.

### 4.4. Evaluation of evolved patterns

We evaluated the best augmented patterns (table 5) as a separate classification model. During aggregations, similarly to the dictionary predictions, positive predictions of the pattern model are assigned a probability of 100%. Table 6 shows evaluation results. As a first observation, classifier ensembles trained both on gold-standard and silver annotation data do not perform better than classifier ensembles trained on gold-standard data, only. Combining the dictionary and the pattern model compensates for the lack of a lower-quality model both for the MaxEnt and the perceptron classifier. Comparing tables 2, 4 and 6 demonstrates how we gradually moved from the recall range 84%-88% to 89%-93%, while keeping precision above 96%-97%. In fact, there are some verified annotation inconsistencies in the test corpus responsible for a minor decrease in precision. More specifically, some terms, such as *3-hydroxyquinidine*, *cycloguanil* and *4-hydroxyomeprazole*, have not

been appropriately tagged as drugs.

### 4.5. Ignoring gold-standard data

In our experiments so far, we assumed that at least some gold-standard data is available for training. However this might not always be the case. In this section, we are concerned with the question: *"How much worse would results be, in the absence of a gold-standard training set?"* This is an important question because, as discussed earlier, gold-standard annotations are time consuming and costly. Ignoring expensive annotations, we experiment with classifiers trained on the easy-to-produce automatically generated annotations, the dictionary and the pattern model. The same gold-standard corpus was used for testing and each incremental improvement was also tested for statistical significance against the previous one using chi-square test, with and without Yate's correction. In all cases, improvements were found to be statistically significant with p-values ranging from $10^{-4}$ to $4 \times 10^{-4}$. The results obtained are shown in Table 7.

Comparing these results with the ones from our baseline experiments, presented in Table 2, shows that the MaxEnt classifier trained solely on silver annotation data, combined with the dictionary and the pattern model, achieves similar performance to the MaxEnt classifier trained on gold-standard data. This result is encouraging, since it suggests that access to gold-standard data is not necessarily a prerequisite for high performance drug-NER.

## 5. Discussion

Using a lexical database to annotate NEs in raw text is not a new concept. In fact, since lexical databases are manually annotated, annotating sentences for NEs from scratch certainly contains some level of effort duplication. We attempted to automate the annotation process by utilising such resources. Unfortunately, our results show that using a dictionary as a direct annotator of drug names achieves top precision but limited recall. Classifiers trained on

gold-standard annotations achieved comparable precision but much higher recall.

For these reasons, we attempted to experiment with methods to preprocess DrugBank before using it as an annotator. To increase recall we generalised dictionary entries into regular expression patterns. We were expecting that the patterns would be able to capture drug names that were not listed in the dictionary but share common morphological characteristics, such as suffixes or prefixes, with some dictionary entries.

Obtaining such patterns automatically and accurately is hard and, thus, our list of patterns is neither perfect nor complete. Perhaps a pharmacologist cooperating with a regular-expression expert would find higher quality patterns, i.e. patterns that generalise better. However, we prefer to explore the extent to which automatic methods can address this task adequately. In the future it would be very interesting to compare our automated method with expert-driven regular expressions, together with incorporation of rules derived from existing WHO and FDA drug nomenclature processes.

Throughout our experiments, we relied heavily on the proposed algorithm for aggregating predictions, which is also not perfect. It is based on assumptions that may not hold in a different context. Moreover, the algorithm always favours predictions of the knowledge-based models (dictionaries and regular expressions) against learning models, accepting the inconsistencies of knowledge-based models as valid. The manually constructed dictionary was of major importance for this study, as it was used in a number of ways. It was the basis to extract synonyms, common word-ending patterns, and was also seen as a direct annotator for an entire corpus. Voting systems similar to the proposed prediction aggregation algorithm are becoming increasingly popular mainly to boost performance but also for the overall stability of the resulting classifier [57–61].

It is also noteworthy that both sets of gold-standard data, for training and testing, are of roughly the same size. Contrasting this with other similar NER experiments, we find that the testing-set is usually a lot smaller than the training-set regardless of the evaluation scheme (holdout or cross-validation). This is due to the fact that the problem of data-sparsity is

pervasive across the entire text mining and NLP discipline (with regards to probabilistic training). In practice, this means that there is rarely enough training data, thus splitting it in two equally sized pieces will most likely not lead to satisfying statistics. We decided to leave the data as is, in order for the experiments to be as easily reproducible as possible.

Our results demonstrate that, even though availability of gold-standard data is certainly helpful, it is not a strict requirement with regards to drug NER. Drugs often share several morphological characteristics, which reduces the contextual information that is needed in order to make informed predictions. Nonetheless, it remains to be investigated whether our combination of heterogeneous models will achieve high performance when tested against larger corpora.

## 6. Conclusions and future work

This study mainly focused on achieving high performance drug NER with very limited or no manual annotations. We achieved this by merging predictions from several heterogeneous models including models trained on gold-standard data, models trained on silver annotation data, DrugBank and, finally, evolved regular expression patterns. We have shown that state-of-the-art performance in drug NER is within reach, even in the presence of data sparsity.

Our experiments also show that combining heterogeneous models can achieve similar or comparable classification performance with that of our best performing model trained on gold-standard annotations. We have shown that in the pharmacology domain, static knowledge resources such as dictionaries actually contain more information than is immediately apparent, and therefore can be utilised in other, non-static contexts (i.e. to devise high-precision regular expression patterns). Including synonyms in the dictionary or disambiguating acronyms did not improve results in this study mainly due to certain design decisions that surround the PK corpus. More specifically, none of the tagged acronyms were identified by AcroMine, whereas most of the identified synonyms have simply not been tagged ap-

propriately in the test-set. Generally speaking however, we would expect a significant performance boost from applying these methods.

We plan to extend this work in the future. First of all, we plan to take advantage of all the annotations in the PK corpus. Being able to recognise both drugs and drug-targets is essential for the task of identifying relationships and interactions between them. We are also very interested to see if we can improve on, or find more accurate regular expression patterns in order to enrich our "safety net" model. Moreover, choosing a drug name is a very long and costly process, and therefore generating good quality candidates automatically would be very useful. Finally, we would like to extend our prediction-aggregation algorithm so as to assign probabilities to predictions of the knowledge-based models (dictionaries and regular expressions).

## Acknowledgments

This research was funded by the U.K. EPSRC - Centre for Doctoral Training and by the AHRC-funded "History of Medicine" project (grant number: AH/L00982X/1). It was also facilitated by the Manchester Biomedical Research Centre and the NIHR Greater Manchester Comprehensive Local Research Network. We would like to thank Prof. Garth Cooper for his valuable advice and discussion.

## References

[1] A. M. Cohen and W. R. Hersh, "A survey of current work in biomedical text mining," *Briefings in Bioinformatics*, vol. 6, no. 1, pp. 57–71, 2005.

[2] S. Ananiadou and J. McNaught, *Text Mining for Biology and Biomedicine*. Norwood, MA, USA: Artech House, Inc., 2005.

[3] U.S. Food and Drug Administration, "How FDA reviews proposed drug names." online. URL: `www.fda.gov/downloads/Drugs/DrugSafety/MedicationErrors/ucm080867.pdf` (Accessed: 15 April 2015).

[4] D. S. Wishart, C. Knox, A. C. Guo, D. Cheng, S. Shrivastava, D. Tzur, *et al.*, "DrugBank: a knowledgebase for drugs, drug actions and drug targets," *Nucleic Acids Research*, vol. 36, no. suppl 1, pp. D901–D906, 2008.

[5] A. Subhadarshini, S. Karnik, Z. Wang, S. Philips, J. Duke, S. Quinney, *et al.*, "An integrated pharmacokinetics ontology and semantically annotated corpus for text mining," in *Proceedings of Summit on Translational Bioinformatics*, (Bethesda, MD, USA), American Medical Informatics Association (AMIA), Mar. 2012.

[6] Linguistic Data Consortium, "ACE (automatic content extraction) english annotation guidelines for entities." online, June 2008. URL: `www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-entities-guidelines-v6.6.pdf` (Accessed: 15 April 2015).

[7] B. M. Sundheim, "Overview of results of the MUC-6 evaluation," in *MUC6 '95: Proceedings of the 6th Conference on Message Understanding*, MUC6 '95, (Stroudsburg, PA, USA), pp. 13–31, Association for Computational Linguistics, Nov. 1995.

[8] N. A. Chinchor, "MUC-7 named entity task definition," in *Proceedings of the Seventh Message Understanding Conference (MUC-7)* (B. Sundheim, ed.), (Fairfax, VA), Apr. 1998. version 3.5.

[9] E. F. Tjong Kim Sang, "Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition," in *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*, COLING-02, (Stroudsburg, PA, USA), pp. 1–4, Association for Computational Linguistics, 2002.

[10] E. F. Tjong Kim Sang and F. De Meulder, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," in *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, (Stroudsburg, PA, USA), pp. 142–147, Association for Computational Linguistics, 2003.

[11] F. Liu, Y. Chen, and B. Manderick, "Named entity recognition in biomedical literature: A comparison of support vector machines and conditional random fields," in *ICEIS (Selected Papers)* (J. Filipe, J. Cordeiro, and J. Cardoso, eds.), vol. 12 of *Lecture Notes in Business Information Processing*, (Berlin & Heidelberg, Germany), pp. 137–147, Springer, 2007.

[12] R. R. V. Goulart, V. L. S. de Lima, and C. C. Xavier, "A systematic review of named entity recognition in biomedical texts," *Journal of the Brazilian Computer Society*, vol. 17, no. 2, pp. 103–116, 2011.

[13] S. Doan and H. Xu, "Recognizing medication related entities in hospital discharge summaries using support vector machine," in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters* (C.-R. Huang and D. Jurafsky, eds.), COLING '10, (Stroudsburg, PA, USA), pp. 259–266, Association for Computational Linguistics, 2010.

[14] D. Sanchez-Cisneros and F. Aparicio Gali, "UEM-UC3M: An ontology-based named entity recognition system for biomedical texts," in *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)* (S. Manandhar and D. Yuret, eds.), (Atlanta, Georgia, USA), pp. 622–627, Association for Computational Linguistics, June 2013.

[15] J. Björne, S. Kaewphan, and T. Salakoski, "UTurku: Drug named entity recognition and drug-drug interaction extraction using svm classification and domain knowledge," in *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)* (S. Manandhar and D. Yuret, eds.), (Atlanta, Georgia, USA), pp. 651–659, Association for Computational Linguistics, June 2013.

[16] I. Segura-Bedmar, P. Martínez, and M. Herrero Zazo, "Semeval-2013 task 9: Extraction of drug-drug interactions from biomedical texts

(ddiextraction 2013)," in *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)* (S. Manandhar and D. Yuret, eds.), (Atlanta, Georgia, USA), pp. 341–350, Association for Computational Linguistics, June 2013.

[17] D. Rebholz-Schuhmann, A. J. J. Yepes, E. M. van Mulligen, N. Kang, J. Kors, D. Milward, *et al.*, "The CALBC silver standard corpus - harmonizing multiple semantic annotations in a large biomedical corpus," in *Proceedings of the Third International Symposium on Languages in Biology and Medicine*, LBM 2009, pp. 64–72, Nov. 2009.

[18] D. Rebholz-Schuhmann, M. Arregui, S. Gaudan, H. Kirsch, and A. Jimeno, "Text processing through web services: calling Whatizit," *Bioinformatics*, vol. 24, no. 2, pp. 296–298, 2008.

[19] M. J. Schuemie, R. Jelier, and J. A. Kors, "Peregrine: Lightweight gene name normalization by dictionary lookup," in *Proceedings of the Second BioCreative Challenge Evaluation Workshop*, BioCreative II, (Madrid, Spain), pp. 131–133, CNIO Centro Nacional de Investigaciones Oncológicas, Carlos III, 2007.

[20] J. Wermter, K. Tomanek, and U. Hahn, "High-performance gene name normalization with GeNo," *Bioinformatics*, vol. 25, no. 6, pp. 815–821, 2009.

[21] A. R. Aronson and F.-M. Lang, "An overview of MetaMap: historical perspective and recent advances," *Journal of the American Medical Informatics Association*, vol. 17, no. 3, pp. 229–236, 2010.

[22] D. Milward and P. Milligan, "Text data mining using interactive information extraction," in *Proceedings of the BioLink Special Interest Group on Text Mining: Linking Literature, Information and Knowledge for Biology*, BioLinkSIG 2007, 2007.

[23] D. Rebholz-Schuhmann, A. Jimeno-Yepes, C. Li, S. Kafkas, I. Lewin, N. Kang, *et al.*, "Assessment of NER solutions against the first

and second CALBC silver standard corpus," in *Semantic Mining in Biomedicine* (N. Collier, U. Hahn, D. Rebholz-Schuhmann, F. Rinaldi, and S. Pyysalo, eds.), vol. 714 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2010.

[24] Y. Tsuruoka, J. McNaught, J. Tsujii, and S. Ananiadou, "Learning string similarity measures for gene/protein name dictionary look-up using logistic regression," *Bioinformatics*, vol. 23, no. 20, pp. 2768–2774, 2007.

[25] C. Kolářik, M. Hofmann-Apitius, M. Zimmermann, and J. Fluck, "Identification of new drug classification terms in textual resources," *Bioinformatics*, vol. 23, no. 13, pp. i264–i272, 2007.

[26] M. A. Hearst, "Automatic acquisition of hyponyms from large text corpora," in *Proceedings of the 15th International Conference on Computational Linguistics - Volume 2*, COLING '92, (Stroudsburg, PA, USA), pp. 539–545, Association for Computational Linguistics, 1992.

[27] E. S. Chen, G. Hripcsak, H. Xu, M. Markatou, and C. Friedman, "Automated acquisition of disease drug knowledge from biomedical and clinical documents: an initial study," *Journal of the American Medical Informatics Association (JAMIA)*, vol. 15, pp. 87–98, Feb. 2008.

[28] C. Feng, F. Yamashita, and M. Hashida, "Automated extraction of information from the literature on chemical-CYP3A4 interactions," *Journal of Chemical Information and Modeling*, vol. 47, no. 6, pp. 2449–2455, 2007.

[29] K. M. Hettne, R. H. Stierum, M. J. Schuemie, P. J. M. Hendriksen, B. J. A. Schijvenaars, E. M. v. Mulligen, *et al.*, "A dictionary to identify small molecules and drugs in free text," *Bioinformatics*, vol. 25, pp. 2983–2991, Nov. 2009.

[30] T. Rindflesch, L. Tanabe, J. N. Weinstein, and L. Hunter, "EDGAR: Extraction of drugs, genes and relations from the biomedical literature,"

in *Pacific Symposium of Biocomputing*, pp. 514–525, 2000. Available online at: psb.stanford.edu/psb-online/proceedings/psb00/ (Accessed on: 15 April 2015).

[31] Y. Usami, H.-C. Cho, N. Okazaki, and J. Tsujii, "Automatic acquisition of huge training data for bio-medical named entity recognition," in *Proceedings of BioNLP 2011 Workshop* (K. B. Cohen, D. Demner-Fushman, S. Ananiadou, J. Pestian, J. Tsujii, and B. Webber, eds.), BioNLP '11, (Stroudsburg, PA, USA), Association for Computational Linguistics, 2011.

[32] A. Vlachos and C. Gasperin, "Bootstrapping and evaluating named entity recognition in the biomedical domain," in *Proceedings of the HLT-NAACL BioNLP Workshop on Linking Natural Language and Biology* (K. Verspoor, K. B. Cohen, B. Goertzel, and I. Mani, eds.), LNLBioNLP '06, (New York, New York), pp. 138–145, Association for Computational Linguistics, 2006.

[33] I. Korkontzelos, A. Vlachos, and I. Lewin, "From gene names to actual genes," in *Proceedings of the BioLink Special Interest Group on Text Mining: Linking Literature, Information and Knowledge for Biology*, BioLinkSIG 2007, 2007.

[34] P. Thomas, T. Bobic, U. Leser, M. Hofmann-Apitius, and R. Klinger, "Weakly labeled corpora as silver standard for drug-drug and protein-protein interaction," in *Proceedings of the Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM) on Language Resources and Evaluation Conference (LREC)* (S. Ananiadou, K. Cohen, D. Demner-Fushman, and P. Thompson, eds.), (Instanbul, Turkey), pp. 63–70, European Language Resources Association (ELRA), May 2012. Available online at: www.lrec-conf.org/proceedings/lrec2012/workshops/14.BioTxtM-Proceedings.pdf#page=70 (Accessed on: 15 April 2015).

[35] S. Kerrien, B. Aranda, L. Breuza, A. Bridge, F. Broackes-Carter, C. Chen, *et al.*, "The IntAct molecular interaction database in 2012," *Nucleic Acids Research*, vol. 40, no. D1, pp. D841–D846, 2012.

[36] I. Dagan and S. P. Engelson, "Committee-based sampling for training probabilistic classifiers," in *Proceedings of the Twelfth International Conference on Machine Learning* (A. Prieditis and S. J. Russell, eds.), The Morgan Kaufmann series in machine learning, (San Francisco, CA, USA), pp. 150–157, Morgan Kaufmann Publishers Inc., 1995.

[37] C. A. Thompson, M. E. Califf, and R. J. Mooney, "Active learning for natural language parsing and information extraction," in *Proceedings of the Sixteenth International Conference on Machine Learning* (I. Bratko and S. Džeroski, eds.), The Morgan Kaufmann series in machine learning, (San Francisco, CA, USA), pp. 406–414, Morgan Kaufmann Publishers Inc., 1999.

[38] K. Tomanek, J. Wermter, and U. Hahn, "An approach to text corpus construction which cuts annotation costs and maintains reusability of annotated data," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)* (J. Eisner, ed.), (Prague, Czech Republic), pp. 486–495, Association for Computational Linguistics, June 2007.

[39] Y. Tsuruoka, J. Tsujii, and S. Ananiadou, "Accelerating the annotation of sparse named entities by dynamic sentence selection," in *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing* (D. Demner-Fushman, S. Ananiadou, K. B. Cohen, J. Pestian, J. Tsujii, and B. Webber, eds.), BioNLP '08, (Columbus, Ohio, USA), pp. 30–37, Association for Computational Linguistics, June 2008.

[40] A. L. Berger, V. J. D. Pietra, and S. A. D. Pietra, "A maximum entropy approach to natural language processing," *Computational Linguistics*, vol. 22, pp. 39–71, Mar. 1996.

[41] F. Rosenblatt, "The perceptron: A perceiving and recognizing automaton," Report 85-460-1, Project PARA, Cornell Aeronautical Laboratory, Ithaca, New York, Jan. 1957.

[42] D. Radev, S. Teufel, H. Saggion, W. Lam, J. Blitzer, H. Qi, *et al.*, "Evaluation challenges in large-scale document summarization," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, (Sapporo, Japan), pp. 375–382, Association for Computational Linguistics, July 2003.

[43] C. Mihăilă and R. T. B. Batista-Navarro, "What's in a name? entity type variation across two biomedical subdomains," in *Proceedings of the Student Research Workshop at the 13th Conference of the European Chapter of the Association for Computational Linguistics* (P. Lison, M. Nilsson, and M. Recasens, eds.), (Avignon, France), pp. 38–45, Association for Computational Linguistics, Apr. 2012.

[44] I. Segura-Bedmar, P. Martínez, and C. De Pablo-Sánchez, "Using a shallow linguistic kernel for drug-drug interaction extraction," *Journal of Biomedical Informatics*, vol. 44, pp. 789–804, Oct. 2011.

[45] N. Okazaki, S. Ananiadou, and J. Tsujii, "Building a high quality sense inventory for improved abbreviation disambiguation," *Bioinformatics*, vol. 26, no. 9, pp. 1246–1253, 2010.

[46] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, ACL '96, (Stroudsburg, PA, USA), pp. 310–318, Association for Computational Linguistics, 1996.

[47] F. Jelinek, B. Merialdo, S. Roukos, and M. J. Strauss, "Self-organized language modeling for speech recognition," in *Readings in Speech Recognition* (A. Waibel and K.-F. Lee, eds.), (San Francisco, CA, USA), pp. 450–506, Morgan Kaufmann, 1990.

[48] S. M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," in *IEEE Transactions on Acoustics, Speech and Singal processing*, vol. ASSP-35, pp. 400–401, Mar. 1987.

[49] M. Collins and J. Brooks, "Prepositional phrase attachment through a backed-off model," in *Proceedings of the Third Workshop on Very Large Corpora* (D. Yarowsky and K. Church, eds.), (Cambridge, MA, USA), pp. 27–38, Association for Computational Linguistics, 1995.

[50] D. Roth and D. Zelenko, "Part of speech tagging using a network of linear separators," in *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 2*, ACL '98, (Montreal, Quebec, Canada), pp. 1136–1142, Association for Computational Linguistics, 1998.

[51] D. M. Bikel, R. Schwartz, and R. M. Weischedel, "An algorithm that learns what's in a name," *Machine Learning*, vol. 34, pp. 211–231, Feb. 1999.

[52] G. Zhou, J. Su, and L. Yang, "Resolution of data sparseness in named entity recognition using hierarchical features and feature relaxation principle," in *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing* (A. Gelbukh, ed.), CICLing'05, (Berlin & Heidelberg, Germany), pp. 750–761, Springer-Verlag, 2005.

[53] G. Campbell and J. E. Mosimann, "Multivariate methods for proportional shape," in *Proceedings of the Section on Statistical Graphics*, Annual Meeting of the American Statistical Association, (Alexandria, VA, USA), pp. 10–17, American Statistical Association, 1987.

[54] R. H. Hijazi, *Analysis of compositional data using Dirichlet covariate models*. PhD thesis, The American University, Washington, DC, USA, 2003.

[55] R. A. Hijazi and R. W. Jernigan, "Modelling compositional data using dirichlet regression models," *Journal of Applied Probability and Statistics*, vol. 4, no. 1, pp. 77–91, 2009.

[56] J. R. Koza, F. H. Bennett, D. Andre, and M. A. Keane, *Genetic Programming III: Darwinian Invention and Problem Solving*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1st ed., May 1999.

[57] P. N. Bennett, S. T. Dumais, and E. Horvitz, "The combination of text classifiers using reliability indicators," *Information Retrieval*, vol. 8, pp. 67–100, Jan. 2005.

[58] K. Al-Kofahi, A. Tyrrell, A. Vachher, T. Travers, and P. Jackson, "Combining multiple classifiers for text categorization," in *Proceedings of the tenth international conference on Information and knowledge management*, CIKM '01, (New York, NY, USA), pp. 97–104, ACM, 2001.

[59] Y. Yang, T. Ault, and T. Pierce, "Combining multiple learning strategies for effective cross-validation," in *International Conference on Machine Learning* (J. Furnkranz and T. Joachims, eds.), (Madison, WI, USA), pp. 1167–1174, The International Machine Learning Society, 2000.

[60] Y. Bi, S. Mcclean, and T. Anderson, "Combining rough decisions for intelligent text mining using Dempster's rule," *Artificial Intelligence Review*, vol. 26, pp. 191–209, Nov. 2006.

[61] L. Si, "Boosting performance of bio-entity recognition by combining results from multiple systems," in *Proceedings of the 5th International Workshop on Bioinformatics*, BIOKDD '05, (New York, NY, USA), pp. 76–83, ACM Press, 2005.

[62] R. Poli, W. B. Langdon, and N. F. McPhee, *A Field Guide to Genetic Programming*. Published via `lulu.com` and freely available at: `www.gp-field-guide.org.uk` (Accessed: 15 April 2015), 2008. (With contributions by J. R. Koza).

# Algorithms

**Algorithm 1** Aggregation of predictions

```
List L →[ ]
for all Sentence s ∈ Text do
    Map M → {:prediction :confidence}
    for all Tokens t ∈ s do
        if ∃ dictionary then
            if dictionary.predict(t) → POSITIVE then
                PUT M {prediction 1.0}
            else
                for all Model m ∈ Models do
                    if m.predict(t) → POSITIVE then
                        STORE {prediction confidence}
                    end if
                end for
                PUT M {prediction max-confidence}
            end if
        end if
    end for
    DROP overlapping/intersecting spans*
    ADD L M
end for
return L
```

*** Rules for dropping spans:**

- *Identical/Intersecting*: first span is kept
- *Contained*: Contained spans are dropped

# Figures

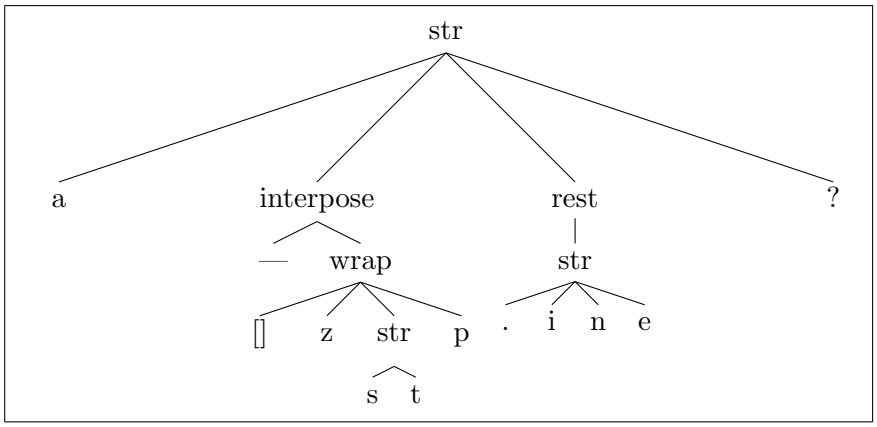


Figure 1: The best-evolved "organism"

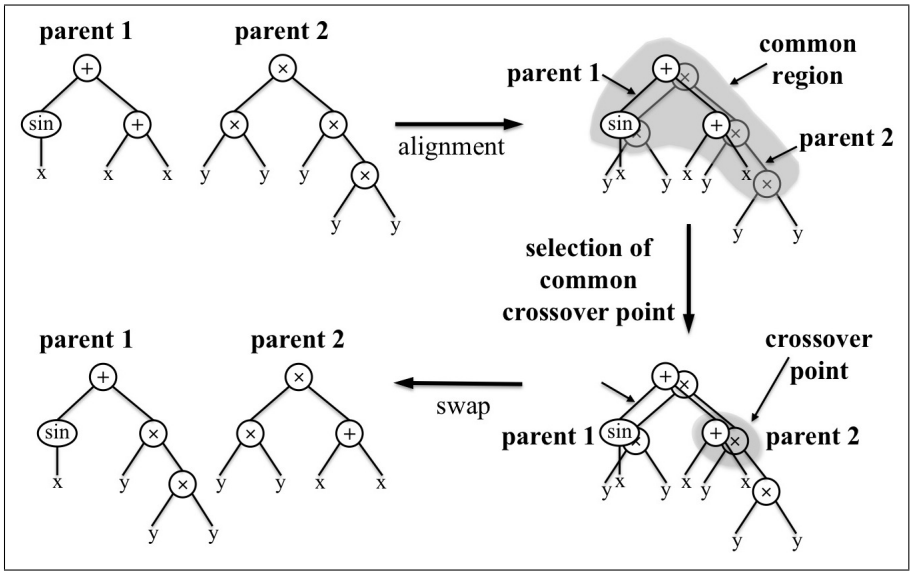


Figure 2: Example of one-point crossover between parents of different sizes and shapes. Image source: [62]

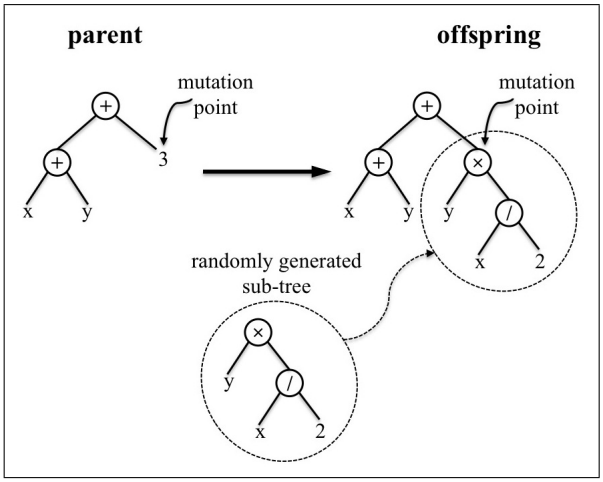


Figure 3: Example of subtree mutation. Image source: [62]

## Tables

| token type features | |
|---|---|
| initial_capital_letter | contains_hyphen |
| all_lowercase_letters | contains_slash |
| all_letters | contains_period |
| all_digits | contains_uppercase |
| contains_digit | contains_letters |

Table 1: Binary token type features

| **classifier** | **P** | **R** | **$F_1$** |
|---|---|---|---|
| Dictionary | **99.7%** | 78.5% | 87.8% |
| Dictionary + synonyms | 93.4% | 78.9% | 85.6% |
| MaxEnt(gold) | 98.3% | 84.5% | 91.0% |
| Perceptron(gold) | 97.5% | 72.0% | 82.8% |
| MaxEnt(gold) + Dictionary | 99.1% | **88.4%** | **93.4%** |
| Perceptron(gold) + Dictionary | 97.6% | 84.3% | 90.4% |

Table 2: Results of baseline classifiers trained on gold-standard data (P: precision, R: recall, $F_1$: F-Score)

| **category** | general drug names | branded drug names |
|---|---|---|
| **system** | NLM_LHC [16] | UTurku [15] |
| **team** | National Library of Medicine | University of Turku |
| **approach** | dictionary-based | SVM classifier (TEES) |
| **precision** | 72.5% | 94.5% |
| **recall** | 91.7% | 88.1% |
| **F-Score** | 81.0% | 91.2% |

Table 3: Results of the best performing systems (in terms of exact-matching F-Score) in general and branded drug name recognition in the DDIExtraction 2013 task [16].

| **classifier** | **P** | **R** | **$F_1$** |
|---|---|---|---|
| MaxEnt (silver) | 98.7% | 47.1% | 63.7% |
| MaxEnt (silver) + dictionary | 99.2% | 78.5% | 87.6% |
| Perceptron (silver) | 98.3% | 76.9% | 86.3% |
| Perceptron (silver) + dictionary | **99.3%** | 78.5% | 87.7% |
| MaxEnt (gold + silver) | 97.8% | 84.8% | 91.0% |
| MaxEnt (gold + silver) + dictionary | 98.6% | **89.7%** | **93.9%** |
| Perceptron (gold + silver) | 97.2% | 79.1% | 87.2% |
| Perceptron (gold + silver) + dictionary | 98.0% | 85.1% | 91.1% |

Table 4: Results of classifiers trained on gold-standard and silver annotation data (P: precision, R: recall, $F_1$: F-Score)

| **evolved patterns** | **matches** | **example** |
|---|---|---|
| a(z\|st\|p)ine? | 130 | nevirapine |
| (i\|u)dine? | 72 | lepirudin |
| azo(l\|n)e? | 62 | fluconazole |
| tamine? | 44 | dobutamine |
| zepam | 17 | bromazepam |
| zolam | 13 | haloxazolam |
| (y\|u)lline? | 12 | enprofylline |
| artane? | 11 | eprosartan |
| retine? | 10 | hesperetin |
| navir | 9 | saquinavir |
| ocaine | 9 | benzocaine |

Table 5: Evolved patterns

| **classifier** | **P** | **R** | **$F_1$** |
|---|---|---|---|
| MaxEnt (gold) + dictionary + patterns | 97.3% | 93.0% | 95.1% |
| MaxEnt (gold + silver) + dictionary + patterns | 97.3% | 93.0% | 95.1% |
| Perceptron (gold) + dictionary + patterns | 95.8% | 88.9% | 92.3% |
| Perceptron (gold + silver) + dictionary + patterns | 96.0% | 88.8% | 92.3% |

Table 6: Evaluation results of ensembles that contain the pattern classifier (P: precision, R: recall, $F_1$: F-Score)

| **classifier** | **P** | **R** | **$F_1$** |
|---|---|---|---|
| MaxEnt (silver) + dictionary + patterns | 97.4% | 85.4% | 91.0% |
| Perceptron (silver) + dictionary + patterns | 97.3% | 85.1% | 90.8% |

Table 7: Results of classifiers that did not use gold-standard data (P: precision, R: recall, $F_1$: F-Score)