LeadMine: A grammar and dictionary driven approach to chemical entity recognition

Daniel M. Lowe[*1] and Roger A. Sayle[2]

NextMove Software Ltd, Innovation Centre, Unit 23, Science Park, Milton Road, Cambridge, United Kingdom

[*1]daniel@nextmovesoftware.com;
[2]roger@nextmovesoftware.com

Abstract. We present a system employing large grammars and dictionaries to recognize a broad range of chemical entities. The system utilizes these resources to identify chemical entities without an explicit tokenization step. To allow recognition of terms slightly outside the coverage of these resources we employ spelling correction, entity extension, and merging of adjacent entities. Recall is enhanced by the use of abbreviation detection and precision is enhanced by the removal of abbreviations of non-entities. With the use of training data to produce further dictionaries of terms to recognize/ignore our system achieved 86.2% precision and 85.0% recall on an unused development set.

Keywords. LeadMine; grammars; abbreviation detection; entity extension

1 Introduction

BioCreative is a series of challenges that has traditionally focused on the recognition and handling of biochemical entities. BioCreative IV introduces the Chemical compound and drug name recognition task (CHEMDNER)[1] which instead is focused on identifying chemical entities. Chemical entity recognition is important for identifying relevant documents and in text mining efforts to extract relationships involving chemicals e.g. drug-disease.

Due to their diversity, quantity and availability, PubMed abstracts were chosen as the corpus for this exercise with 10,000 being manually annotated by the BioCreative team. 7,000 were provided to participants (divided equally into training and development sets) whilst the unseen 3,000 were used to evaluated the performance of the solutions.

Attempts to tackle the problem of chemical entity recognition have invariably identified that the problem is not amenable to pure dictionary approaches due to the continuing discovery of novel compounds and the many ways in which systematic nomenclature allows compounds to be named[2]. Hence, state of the art systems use machine learning techniques to extrapolate from training data the features that are associated with chemical nomenclature. Examples include OSCAR4 which employs a maximum-entropy Markov model[3] and ChemSpot which employs a conditional random field model[4]. Comprehensive reviews of the area have been performed by Vazquez et al.[5] and Gurulingappa et al.[6]

LeadMine instead takes the approach of attempting to encode the rules used to describe systematic chemical nomenclature (as grammars), with large dictionaries being used for trivial names.

As compared to machine learning approaches this makes the results readily understandable; false positives can be pin-pointed to a particular grammar/dictionary and false negatives are readily correctable by adding the relevant nomenclature rule to a grammar or trivial name to a dictionary.

2 Discussion

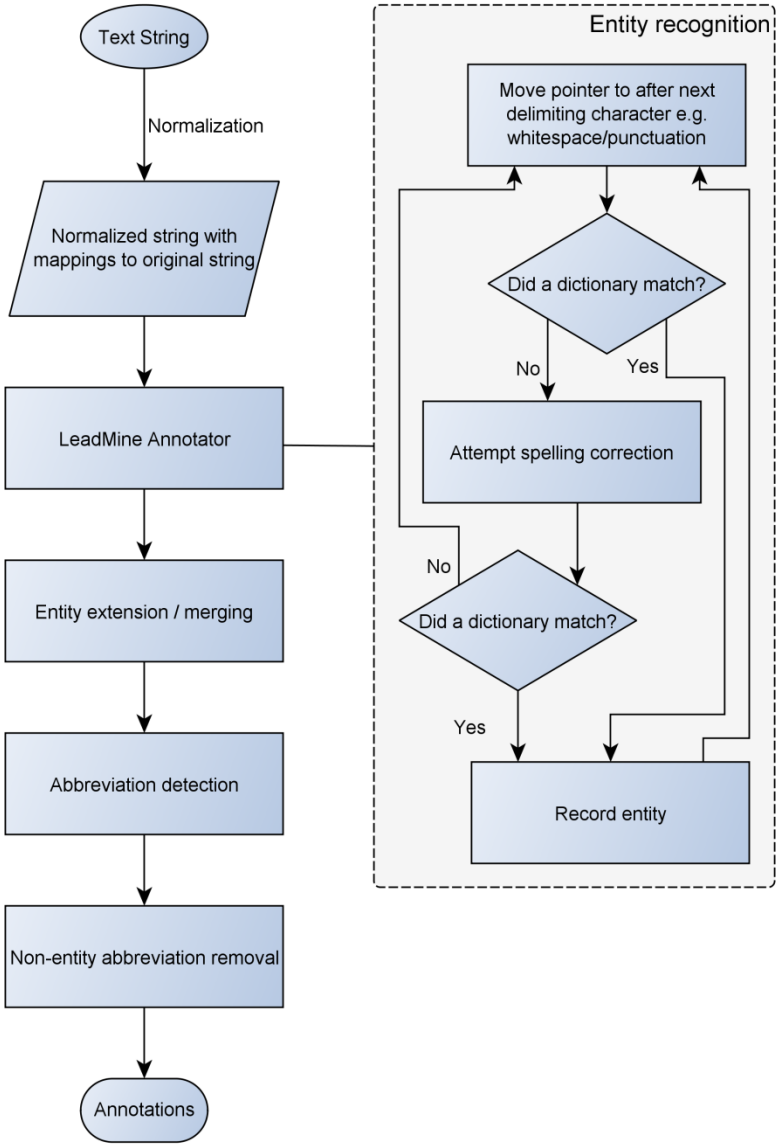

Figure 1 Annotation workflow diagram

Figure 1 shows the workflow we developed; the steps are expounded on below. It should be noted that every step after the LeadMine Annotator can be considered a form of post-processing, and any or all of these steps may be omitted.

2.1 Normalization

To address the issue of there being many Unicode forms for characters with similar meaning a normalization step is performed. For example `(backtick), ‘ and ’ (single quotation marks) and ′ (prime) are all converted to apostrophe. Another example is œ which is converted to oe. This step reduces the number of trivial variants that dictionaries/grammars need to match. The normalization step also facilitates processing of XML documents by removing all tags. For example, `<p>H<sub>2</sub>O</p>` is normalized to H2O. This ability was, except in a handful of cases, irrelevant to PubMed abstracts. The indexes of characters in the original string are associated with indexes in the normalized string to allow mapping back to the original input.

2.2 LeadMine Annotation

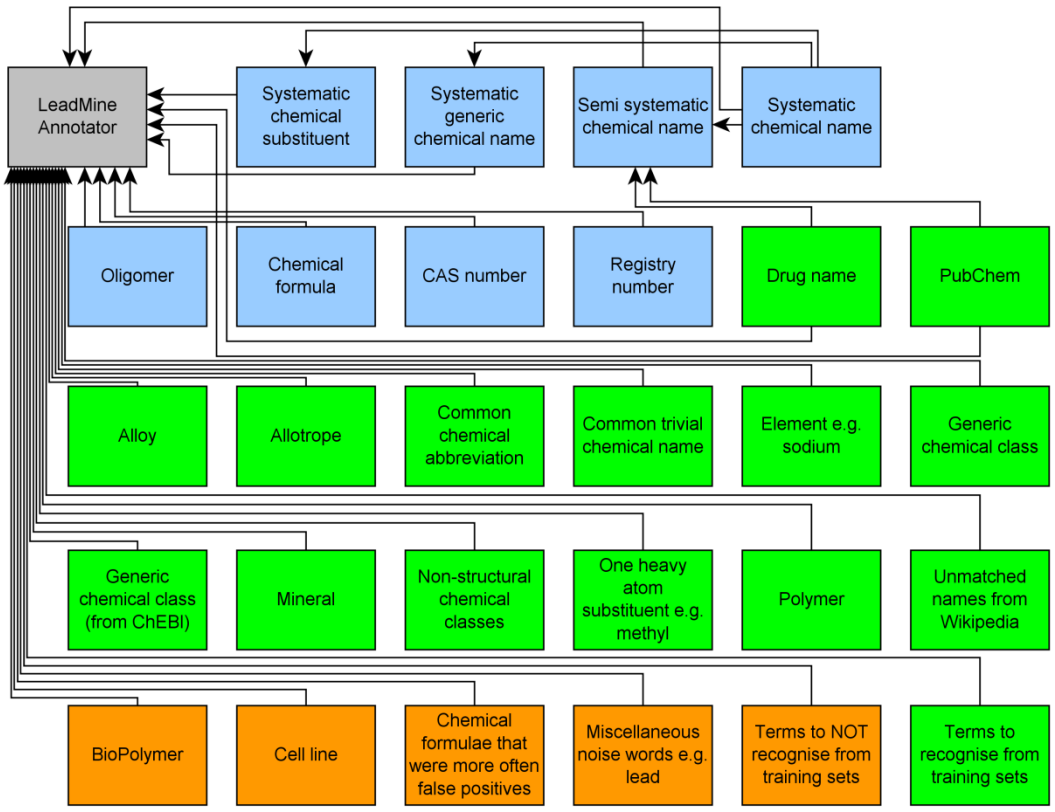

Figure 2 Dictionaries employed by LeadMine for CHEMDNER task (blue: grammars, green: traditional dictionaries, orange: blocking dictionaries)

The rules for chemical nomenclature are encoded as formal grammars e.g.

```
alkanStem : 'meth' | 'eth' | 'prop'…
alkane: alkanStem 'ane'
```

Our grammar for systematic chemical names currently contains 485 of such rules. Grammars may inherit rules from other grammars as shown in Figure 2. To allow conversion of the grammar to an efficient form for matching[7], the rules are restricted to the subset of rules that may be expressed by a regular grammar e.g. a rule may not reference itself. As correct nesting of brackets is not possible with this condition we enforce correct nesting of brackets by keeping track while matching.

The PubChem dictionary is our primary source of trivial names and is 2.94 million terms. It was produced by running a series of filters against the ~94 million synonyms provided by PubChem. Most importantly we removed terms that are an English word or start with an English word. Additionally we inspected the structures present in PubChem as to whether they contained tetrasaccharides (or longer) or hexadecapeptides (or longer) and excluded these records.

2.3 Entity Extension / Merging

We extended entities until we reached whitespace, a mismatched bracket or an English word/noise word. Additionally if an entity was entirely enclosed in balanced brackets and entity extension starting from before/after the brackets yielded a longer entity we used these entity boundaries.

An exception was made for the case of two entities separated by a hyphen where both corresponded to specific compounds. In this case the end and start of the entities respectively are not extended and the entities are not merged. Such a construct often indicates a mixture e.g. ‘Resorcinol-Formaldehyde’.

Next entities are trimmed of “Non-essential parts of the chemical entity and name modifiers” e.g. ‘group’, ‘colloidal’, ‘dye’ etc. Entities that overlap are merged together. Entities that are space separated are merged together unless one of the entities is found to be an instance of the other entity. For example genistein isoflavone is not merged as

genistein IS an isoflavone. These relationships are derived by use of a local copy of the ChEBI[8] ontology.

The aforementioned trimming process is repeated. Finally if after trimming an entity corresponds to a blacklisted term it is excluded. An example is ‘gold nanoparticles’ where ‘nanoparticles’ is excluded by trimming and ‘gold’ is explicitly not to be annotated in the annotation guidelines.

By special case the ‘S’ in glutathione-S-transferase is annotated.

2.4 Abbreviation Detection

We used an adapted version of the Hearst and Schwartz algorithm[9] to identify abbreviations of entities found by the system. By providing the “long form” (unabbreviated) we avoid one of the issues with the algorithm which is that it may not identify the complete unabbreviated form. We extended the algorithm to recognized abbreviations of the following forms:

- Tetrahydrofuran (THF)
- THF (tetrahydrofuran)
- Tetrahydrofuran (THF;
- Tetrahydrofuran (THF,
- (tetrahydrofuran, THF)
- THF = tetrahydrofuran

Abbreviations may contain brackets as long as they are balanced. The conditions described by Hearst and Schwartz are applied with the additional requirements that the short form must not be a common chemical identifier e.g. ‘1a’ or Roman numeral e.g. ‘II’. The minimum length of abbreviations is configurable and set to 3 for compliance with the annotation guidelines.

We also utilize a list of string equivalents to allow, for example, mercury to be abbreviated to Hg. Once an abbreviation has been detected all further instances of that string in that particular abstract are annotated.

2.5 Non-entity abbreviation removal

In this step we postulate that an entity we have discovered is an abbreviation and use Hearst and Schwartz algorithm to find a potential long form for it. If the algorithm finds a suitable long form and this long form is not also an entity or overlaps with an entity we assume that the abbreviation entity is a false positive. We then remove both it and all other instances of it. For example when 'current good manufacturing practice (cGMP)' is seen cGMP clearly doesn't mean cyclic guanosine monophosphate!

3 Evaluation

We used the training set to automatically identify holes in our coverage and identify common false positives and from this derived a dictionary of terms to include (Whitelist) and a dictionary of terms to exclude (BlackList). Below are our results for identifying all chemical entity mentions in the development set.

Configuration	Precision	Recall	F-score
Baseline	0.869	0.820	0.844
WhiteList	0.862	0.850	0.856
BlackList	0.882	0.803	0.841
WhiteList + Blacklist	0.873	0.832	0.852

The addition of a whitelist dictionary provided a significant increase in recall, indicating that there are still gaps in the coverage of the system's dictionaries and grammars. The use of a blacklist dictionary was less successful due to the loss of recall incurred. This is due to the blacklist being formed primarily of genuine chemical entities that in the context of the training set either were not annotated in the gold standard or formed part of longer chemical entities.

The 5 runs submitted for the CEM (chemical entity mentions) and CDI (chemical document indexing) tasks used the following configurations:
Run1: same as Baseline
Run2: same as Whitelist but also used the development set for training
Run3: same as BlackList but also used the development set for training

Run4: same as Whitelist + Blacklist but also used the development set for training
Run5: Same as Whitelist

For the CDI task we used the precision of entities on the development set to predict confidence and hence rank order the entities. We took into account which dictionary was used, whether the entity was in the title or abstract text, whether the entity was extended/merged and whether the entity occurred more than once in the union of the title/abstract text.

REFERENCES

1. Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vazquez, Julen Oyarzabal, Alfonso Valencia. Overview of the chemical compound and drug name recognition (CHEMDNER) task. In: Proceedings of the fourth BioCreative challenge evaluation workshop. Vol. 2. Washington; 2013.

2. Klinger R, Kolarik C, Fluck J, Hofmann-Apitius M, Friedrich CM. Detection of IUPAC and IUPAC-like chemical names. Bioinformatics. 2008;24(13):i268.

3. Jessop DM, Adams S, Willighagen EL, Hawizy L, Murray-Rust P. OSCAR4: a flexible architecture for chemical text-mining. Journal of Cheminformatics. 2011;(3):41.

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

5. Vazquez M, Krallinger M, Leitner F, Valencia A. Text Mining for Drugs and Chemical Compounds: Methods, Tools and Applications. Molecular Informatics. 2011;30(6-7):506–519.

6. Gurulingappa H, Mudi A, Toldo L, Hofmann-Apitius M, Bhate J. Challenges in mining the literature for chemical information. RSC Advances. 2013;3(37):16194–16211.

7. Sayle R, Xie PH, Muresan S. Improved Chemical Text Mining of Patents with Infinite Dictionaries and Automatic Spelling Correction. Journal of Chemical Information and Modeling. 2011;52(1):51–62.

8. Degtyarenko K, de Matos P, Ennis M, Hastings J, Zbinden M, McNaught A, Alcantara R, Darsow M, Guedj M, Ashburner M. ChEBI: a database and ontology for chemical entities of biological interest. Nucleic Acids Research. 2008;36:D344–350.

9. Schwartz A, Hearst M. A Simple Algorithm for Identifying Abbreviation Definitions in Biomedical Text. In: Proceedings of the Pacific Symposium on Biocomputing. Kauai; 2003. pp. 451–462.