Automated exclusion of chemically implausible reactions

Introduction

This paper pertains to techniques that can be used to identify reactions found by text mining that are likely to contain errors from the extraction process or be derived from text that did not in fact contain a reaction. Various automated methods will be documented and evaluated to determine their power to discriminate suspect reactions from correct reactions. The corpus of reactions used for evaluation are those extracted using the Patent Reaction Extraction project[1] from 2001-2012 USPTO applications. The workings of this code are fully described in the related thesis[2].

Reactions data structure

The reactions are output as CML in a hierarchical directory structure:

applications|grants →year→weeklyArchiveName→Patent number→incomplete|complete

weeklyArchiveName is the name of the file as provided by Google's bulk USPTO download service . The "complete" and "incomplete" folders contain those reactions found to be plausible and those that appear to be missing information/not reactions. The algorithm for deciding which folder a putative reaction should be placed in is described below.

A putative reaction must have at least one reactant or product (and these DO NOT need to be name to structure convertible). As the system's primary method of identifying reactions is by finding a heading with a chemical (which can then be implicitly used as a reaction product) it is unusual for a reaction to have 0 products. The 0 product case can arise when a paragraph contains a yield phrase in which no chemical compounds were detected.

Reactions are then created in GGA's Indigo toolkit[3] by loading the SMILES of the individual components (where name to structure succeeded). All the following checks apply to Indigo's representation of the reaction.

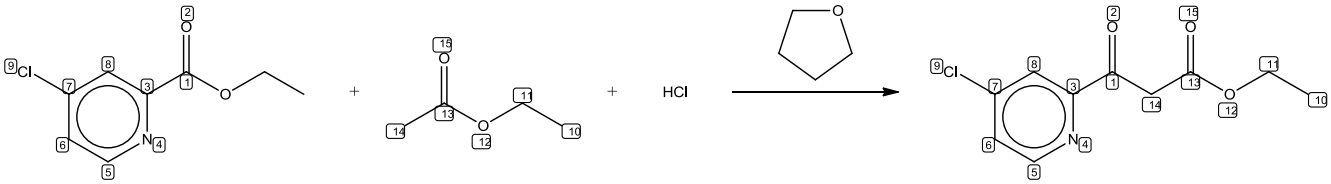
Reactions meeting the following criteria are immediately classified as incomplete:

- Products = 0
- Reagents (reactants + agents) < 2
- All products are reagents (checked by InChI comparison, the grouping of components is taken into account so [pyridine] and [hydrochloric acid] is not the same as [pyridine hydrochloride])

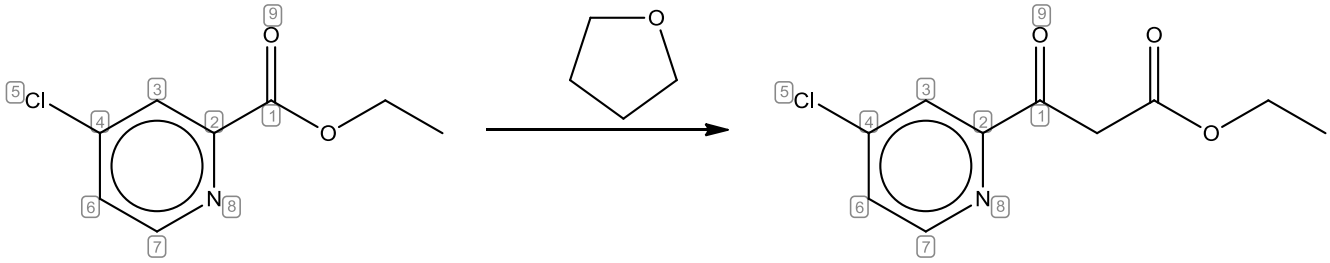
As the system was designed around being high recall with filtering used to obtain the required precision it is to be expected that most incomplete reactions are false positives.

The remaining reactions are then atom-atom mapped by Indigo. If all atoms of the product correspond to atoms in the reactants the reaction is classified as complete, otherwise it is incomplete.

Complete reaction:



Incomplete reaction:

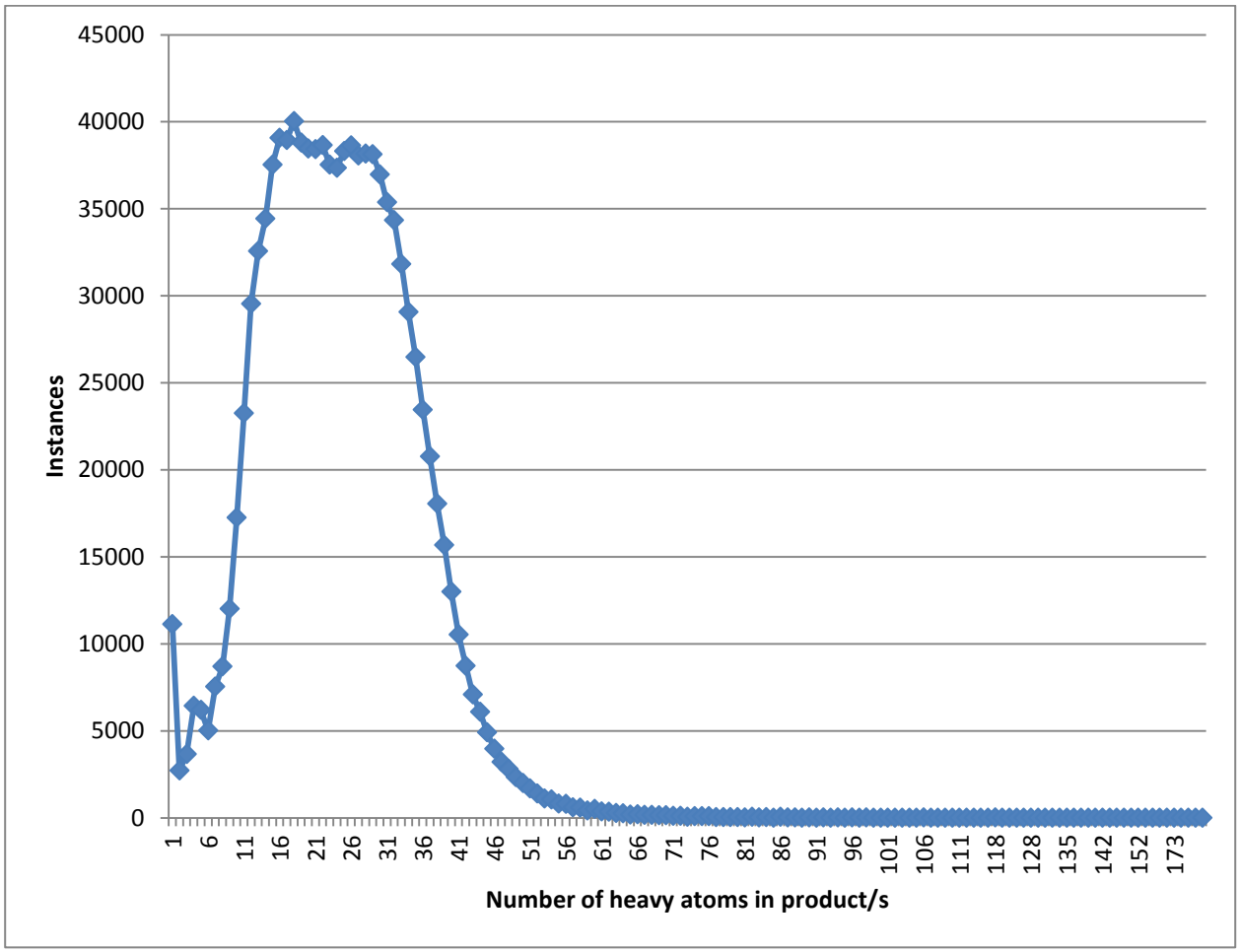


The overall number of reactions categorised as incomplete is approximately the same as the number categorised as complete.

Proposed filtering methods

Unfortunately the methods outlined thus far still leave a significant percent of incorrect reactions. Lowe[2] further filtered the reactions classified as “complete” with filters for unresolvable product components and products that appeared to be non-specific or fragments. On an evaluation of 100 reactions this gave 95% precision for reactions with both the correct product and primary starting material. Hence one would still expect on a dataset of a million reactions to encounter 50,000 erroneous reactions! The two proposed filters as well as other novel filters are described and evaluated in this section.

Product heavy atom count filter



The distribution of product sizes suggests that there are more very small products than might be expected with one atom products representing the clearest outlier. A one atom product can erroneously arise from phrases such as “to yield the hydrochloride salt”.

Method

Sum up heavy atoms in all products of a reaction. Reject reaction based on a minimum product size threshold.

Product charge filter

The products of a chemical reaction will nearly always be overall neutral. Whilst a charge unbalanced product could arise from a missing counter ion (which in itself is an extraction error) another possibility is that the extracted product is the counter ion/salt component and the actual product has been missed. This can arise from phrases such as “to yield the acetate salt”.

Method

In the case where a reaction has 1 product with no disconnected substructures, sum up the charge over all the product atoms. Reject reaction if not equal to 0.

Dubious *definiteReference* name filter

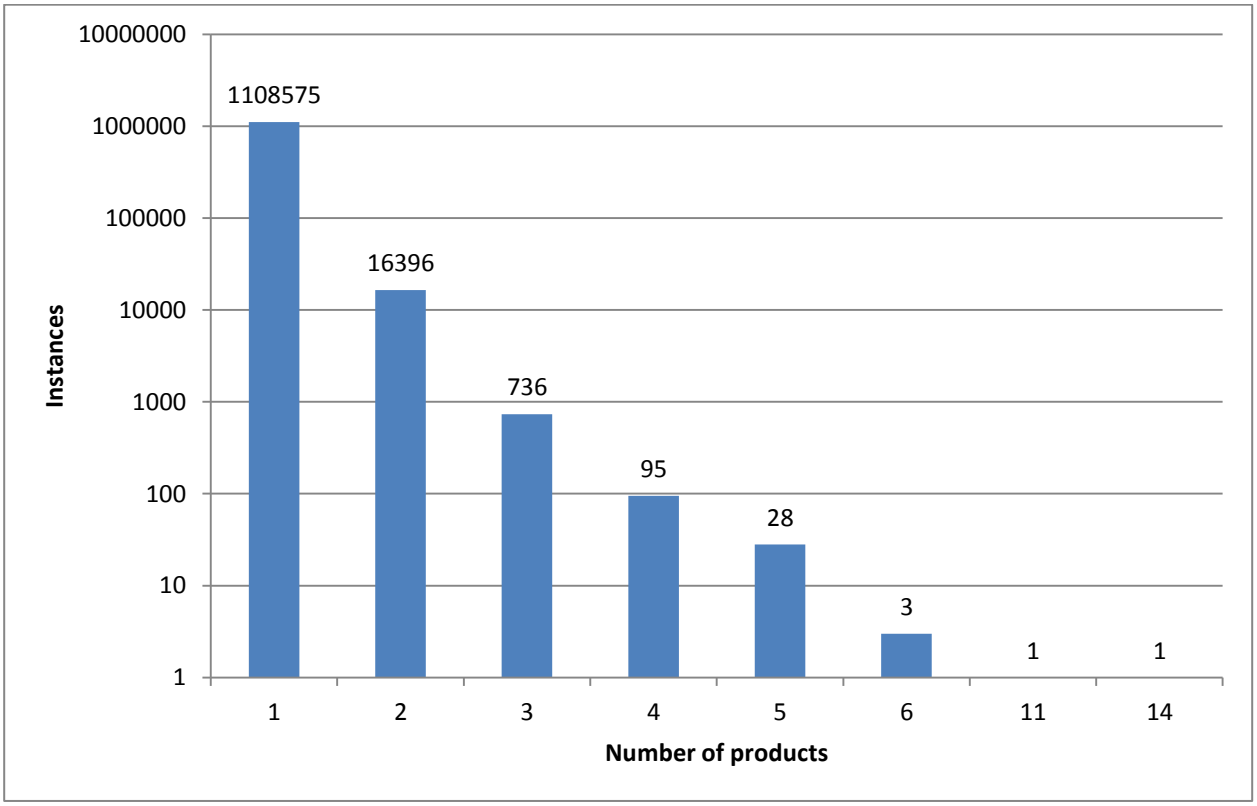
The extraction software categories chemicals as *exact*, *definiteReference*, *chemicalClass* or *fragment*. A *definiteReference* is where a name refers to a specific compound but the name itself is a reference to that compound rather that giving the structure of it. Examples of *definitereference*s include “title compound”, “thiazole (5a)” and “the thiazole”. In the latter two cases where resolution of the name to the specific compound fails thiazole would be erroneously used as the structure. Unfortunately “the” in front of a chemical name is not a perfect indicator that a name is in fact a *definiteReference*. As a result the software currently reclassifies names preceded by “the” as *exact* if the software is unable to resolve the name to a specific compound, the name possesses no further indication that it is a reference (e.g. a numeric identifier) and the name is name to structure convertible.

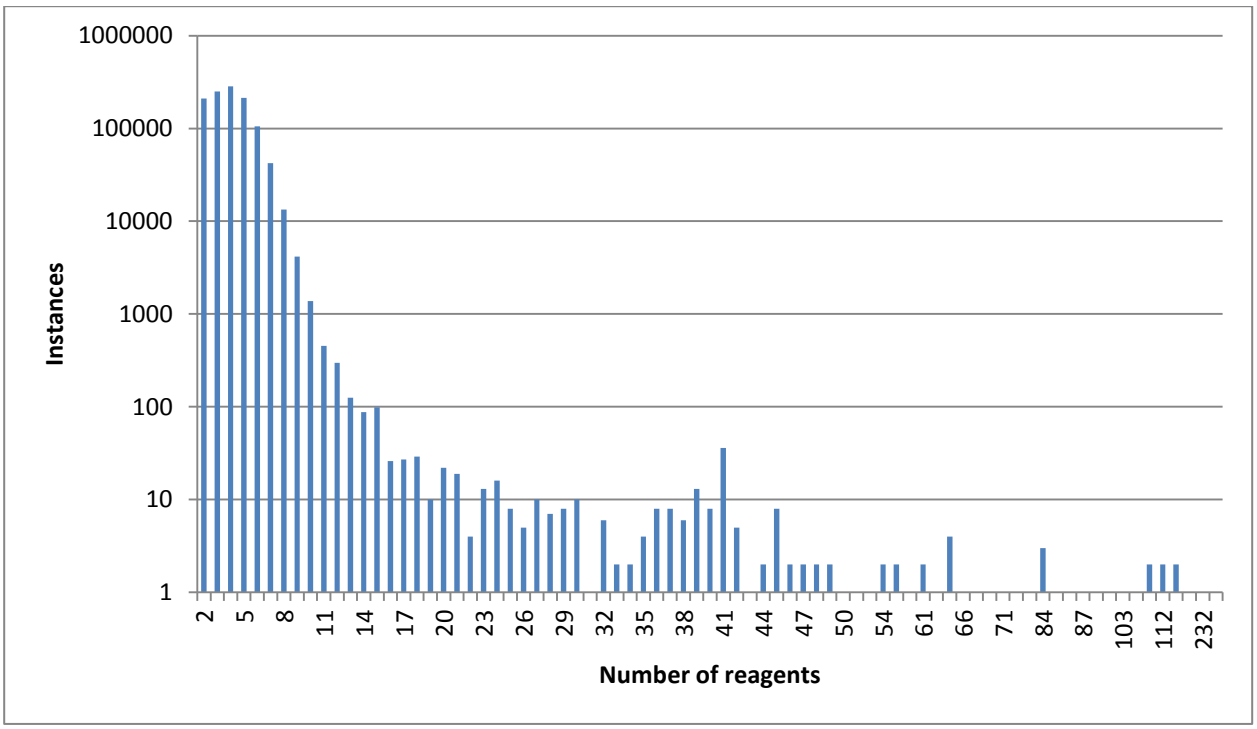
The proposed filter aims to identify cases where a *definiteReference* was not resolved leading to an erroneous structure for a reaction component (often the product). Due to the aforementioned exception where a *definiteReference* may be reclassified back to an *exact* name, some errors caused by this issue will be missed by the filter.

Method

Identify products of type *definiteReference* where the name does not contain a hyphen. Attempt to generate the InChI of the compound from its name. Reject reaction if the generated InChI is equal to the one present in the CML for that compound.

Component count filters





Reaction write-ups typically include a handful of reagents and a single product. As can be seen from graphs above (note the logarithmic scale) this also is found in the patent literature. However there are also a small number of reactions with an excessive number of components which can be expected to be caused by mistakes in the extraction process. Such cases may be problematic for some algorithms e.g. atom-mapping.

Method

Count the number of unique structures associated with reactant/agent components e.g. using InChI. Reject reaction if >15. Count the number of unique structures associated with product component. Reject reaction if >4. NOTE: Components may be formed of disconnected substructures, e.g. sodium chloride is one component. The reaction SMILES in the CML contains a ChemAxon extension which includes the mapping between the disconnected substructures present in a SMILES and the reaction components.

ChemicalClass or *fragment* product filter

Extracted chemical reactions are expected to be those involving specific compounds rather than general reaction schemes. Names of type *chemicalClass* are likely to indicate that this is not a specific reaction. Names of type *fragment* may indicate that a complete chemical name has not been identified.

Method

Query entity type of all products. Reject reaction if entity type equals *chemicalClass* or *fragment*.

Product name cannot be converted to a structure filter

Whilst it is to be expected that some, especially inorganic, reagents will not be convertible to structures; it should be rare that a product cannot be converted to a structure. In the patent literature it is rare for a reaction to have multiple documented products which makes the case where there is a mixture of structure resolvable and unresolvable names even more suspect. The case where a product and its salt are named separately can lead to this situation. If the product is unresolvable this reaction should be rejected.

Method

Search all products for structure identifiers i.e. SMILES/InChI. Reject reaction if any product has 0 identifiers.

Testing methodology

For each proposed filter 20 randomly chosen reactions (from the set of complete reactions) that the filter matched were manually evaluated. If a reaction was incorrect, but for a reason that was unrelated to what was being tested by the filter under consideration then another reaction was randomly chosen. For example if when testing a small product filter, the reactants were incorrect but the product was actually correct.

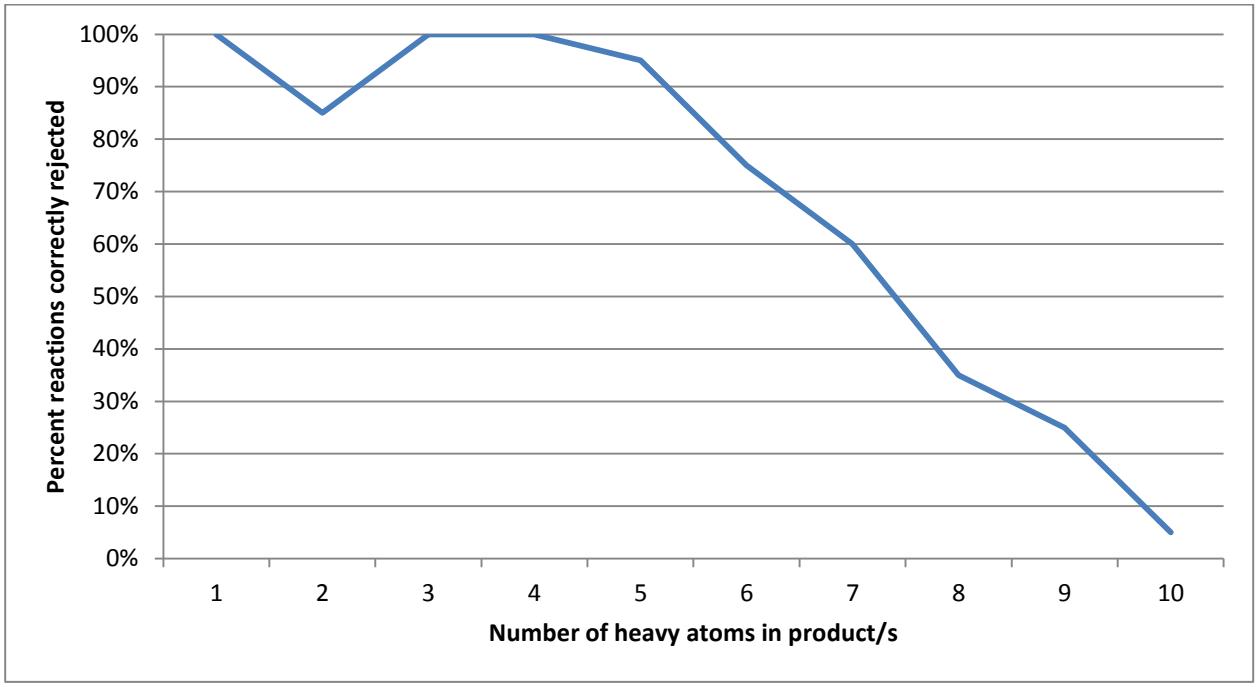
Results

Filter	Hits
Heavy atom count (1 atom product)	11,126
Heavy atom count (2 atom product)	2,706
Heavy atom count (3 atom product)	3,671
Heavy atom count (4 atom product)	6,387
Heavy atom count (5 atom product)	6,212
Heavy atom count (6 atom product)	5,025
Heavy atom count (7 atom product)	7,535
Heavy atom count (8 atom product)	8,640
Heavy atom count (9 atom product)	12,093
Heavy atom count (10 atom product)	17,261
Product charge	22,011
Dubious *definiteReference* name	7,435
Too many products (>4)	33
Too many reagants (>15)	372
Has *chemicalClass* or *fragment* product	18,695
Product name cannot be converted to a structure	48,022

Filter	Precision
Heavy atom count (1 atom product)	20/20 (100%)
Heavy atom count (2 atom product)	17/20 (85%)
Heavy atom count (3 atom product)	20/20 (100%)
Heavy atom count (4 atom product)	20/20 (100%)
Heavy atom count (5 atom product)	19/20 (95%)
Heavy atom count (6 atom product)	15/20 (75%)
Heavy atom count (7 atom product)	12/20 (60%)
Heavy atom count (8 atom product)	7/20 (35%)
Heavy atom count (9 atom product)	5/20 (25%)
Heavy atom count (10 atom product)	1/20 (5%)
Product charge	19/20 (95%)
Dubious *definiteReference* name	20/20 (20%)
Too many products (>4)	17/20 (85%)
Too many reagants (>15)	20/20 (100%)
Has *chemicalClass* or *fragment* product	6/20 (30%)
Product name cannot be converted to a structure	6/20 (30%)

Discussion

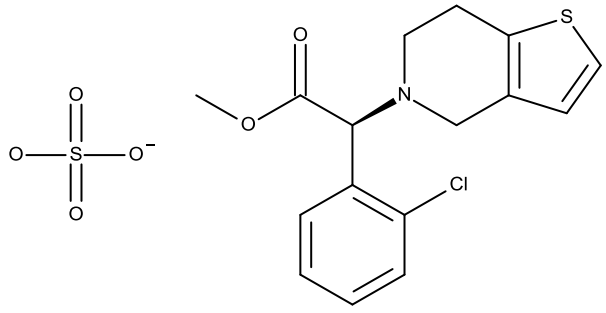
Product heavy atom count filter



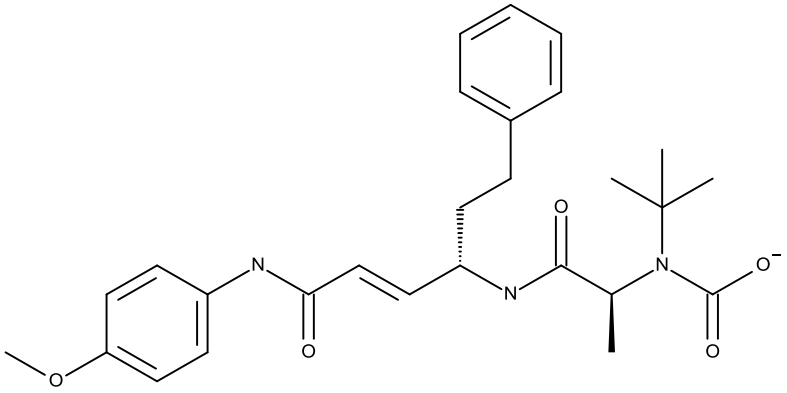
The product heavy atom count filters were able to discriminate well between genuine and erroneous products, at low numbers of heavy atoms, rapidly dropping off as the number of heavy atoms increased. For 10 heavy atoms the precision (5%) is about the same as the overall error rate of the dataset i.e. the filter has no discriminative power.

Product charge filter

The product charge whilst showing up some of the expected salt/generic terms e.g. “the azide”, “as the acetate salt” etc. also revealed issues with the name to structure conversion of certain salts and esters.
When a salt formed of a trivial name and systematic/trivial counter ion is encountered name to structure of the two components is done independently with the structures being subsequently merged. Done without context this often leads to the counterion being charged and the main product being neutral e.g.



As the intended structure is clear this is not considered wrong. To work-around this issue the additional constraint that the product must be formed of 1 fully connected structure for this filter to be applicable was used.



Erroneous, but unambiguous, interpretation of 1,1-dimethylethyl((1S)-1-methyl-2-{[(1S,2E)-4-{[4-(methyloxy)phenyl]amino}-4-oxo-1-(2-phenylethyl)-2-buten-1-yl]amino}-2-oxoethyl)carbamate

OPSIN has heuristics to detect cases where an ester interpretation is intended but the space after the substituent (in this case 1,1,-dimethylethyl) has been omitted. Unfortunately these heuristics are based around whether the name is ambiguous if the non-ester interpretation is assumed. In cases where the "ate group" is small e.g. carbamate or acetate, often both interpretations are unambiguous. In principle more aggressive heuristics that always chose the ester interpretation if not doing so would lead to a charge unbalanced compound, could be employed. Cases such as this were considered genuine errors as the connection table of the product is wrong.

Dubious *definiteReference* name filter

When configured as described this filter achieved perfect precision on subset it was tested on. The requirement that the product must not have a hyphen was added as a heuristic to favour ambiguous names. Without this heuristic it wouldn't be possible to distinguish between "to yield thiazole (5)" and "to yield 2,4-dipropylthiazole (5)" (the latter should ideally be classified as type *exact*).

Component count filters

These filters worked well although their applicability is relatively limited compared to the other filters. Nonetheless the reactions they remove are likely to be the ones that are especially troublesome to processing algorithms. Analysis of the incorrect reactions picked out by the reactant counter filter revealed two common classes of failure both of which can be corrected by small changes to the reaction extraction code. One class of error was caused by tables in older USPTO patents being considered as part of a reaction paragraph; this only occurs on older patents as the schema is different and hence an unanticipated element is used to contain tables. The other class of error is where a paragraph starts with a chemical followed by a semicolon with the rest of the paragraph then being a semicolon delimited list of chemicals. Without the context of what follows the semicolon the reaction extraction code erroneously treats the first name as a sub heading and hence it is the product with the rest of the list being the reactants.

The three reactions which correctly had many products were from reaction descriptions that analysed the percentage of each product of the reaction rather than just the desired product.

ChemicalClass or *fragment* product filter

This filter was found to in practice not be especially precise. One common issue were phrase like “to obtain a solid” where the solid was inferred to be product. The presence of ‘a’ indicates the entity to be of type *chemicalClass*. This could be improved by reclassifying as type *definiteReference* in the case that the entity was implicitly resolved to be the title compound. Using this filter of reactants/agents also has poor performance due to solutions (“ a solution”) and some solvents (“hexanes”) being classified as *chemicalClass*. “hydrogen” is also misclassified as type *fragment* as the program incorrectly expects it to mean a hydrogen atom rather than hydrogen gas.

Product name cannot be converted to a structure filter

This filter was also not found in practice to be especially precise and also matched significantly more reactions than the other filters. A common issue was the description of a product being split into two named entities e.g. 1,2-dichlorobenzene was yielded as a gray solid (5g). “gray solid (5g)” is recognized as a separate entity with no structure. As recognizing these entities as one would obscure the phrase key word (“yielded”) a workable solution might be to attempt to merge the entities as part of the reaction extraction workflow. In cases like this, this would increase the recall of quantities associated with the product.

Other issues included irrelevant chemical entities related to analysis being identified as products, which were then not convertible to structures. The presence of these entities does not change the fact that the correct product was identified.

Conclusions

A significant improvement in the quality of the reaction dataset can be achieved using relatively simple filters. While the filters described here were tested independently, in practice there will be correlation between some of them e.g. many charged products will be very small salts. Hence in principal a model that considered multiple filters simultaneously could perform better. Applying the following filters:

- >8 product heavy atoms
- Product charge filter
- Dubious definiteReference name filter
- productComponentCount <5 and reactantAndAgentComponentCount <16

reduced the 2001-2012 applications data set from 1,125,835 reactions to 1,064,511 reactions. If the datasets are uniquified on reaction SMILES (to eliminate a reaction description being copied between patents) and then the instances of particular products counted: in the unfiltered set 214 products are synthesized 20 or more times whilst in the filtered set only 39 products are synthesized 20 or more times)

References

(1) Lowe, D. M. Patent Reaction Extraction Project https://bitbucket.org/dan2097/patent-reaction-extraction (accessed Jun 4, 2012).

(2) Lowe, D. M. Extraction of chemical structures and reactions from the literature. Thesis, University of Cambridge, 2012.

(3) GGA Software Services Indigo Toolkit http://ggasoftware.com/opensource/indigo (accessed Jun 4, 2012).