

# Efficient synonym search by semantic linking of multiple data sets

Kenny Knecht<sup>1</sup> Bérénice Wulbrecht<sup>1</sup> Filip Pattyn<sup>1</sup> Hans Constant<sup>1</sup>

ONTOFORCE NV, Gent, Belgium,  
kenny@ontoforce.com

**Abstract.** We describe a method to automatically pick a highly relevant subset of synonyms to broaden a text search based on keywords.

Public datasets in the bio-medical area tend to provide a plethora of synonyms or alternative names. It is not uncommon that chemicals or diseases have more than 50 different alternative names in data sets like UMLS or ChEMBL. This may result in inefficient searches and sometimes even in false positives if you use these to extend an initial search. Through semantic linking of several datasets we define a heuristic which increases the power of the search meanwhile making it more efficient. We evaluated the method on the 500 most common keyword searches used the first 6 months of 2017 in the semantic web platform DISCOVER ([www.discover.com](http://www.discover.com)). More than 98% of the hits are retrieved back by submitting only 16% of the synonyms.

We implemented this method as a visual suggestion, which the user can override manually at any time. Notwithstanding the fact that we focus our examples and concrete implementation on the biomedical databases in the publicly available DISCOVER, we would like to stress that the method is much more generally applicable.

**Keywords:** semantic web, text search, synonyms, data integration

## 1 The problem

Traditionally search engines start with a text query. All the documents containing that text are subsequently returned: in DISCOVER this can be publications, clinical trials or funded research programs, but also other concepts like diseases, genes, variants or other chemicals.

The central example will be all documents about the concept *aspirin*. Merely typing "*aspirin*" will surely return a lot of relevant results, but some will also be missed. For example some documents may only mention the more scientific name "*Acetylsalicylic acid*". So logically the user would like to expand the search with synonyms to broaden his or her search, i.e. query for "*aspirin*" OR "*Acetylsalicylic acid*".

Since DISCOVER brings together many data sources -many of them actually contributing synonyms and other alternative names- this can be easily automated. The record for *Aspirin* for example collects data from no less than 13

different databases (HMDB, DrugCentral, DrugBank, HSDB, UNII, ChEMBL, UMLS, ChEBI, IUPHAR Compendium, SureChEMBL, RxNorm, MeSH, PubChem ), from which 8 contribute to alternative names. In total the public databases gives 75 distinct alternative names for *aspirin*, many of which are not strictly synonyms, but hyponyms like "Nu-Seals 300" or "Bayer Extra Strength" (ChEMBL via <http://www.w3.org/2004/02/skos/core#altLabel>). Including all those alternative names in our search will result in a fairly complete result, but may also be very inefficient.

There is also another risk involved. Elaborating on the previous example, one of the alternative names is *ASA*. Although this is used as an alternative name for aspirin, it is also the abbreviation of anti-sarcolemmal autoantibodies Mus Musculus gene, of the disease Argininosuccinic aciduria and many more. A search for this word will inevitably introduce many false positives. Another source of ambiguity may be hypernymy, synonyms that are broader than the actual submitted keyword.

## 2 Related work

While query expansion is a much studied subject we apply it here in highly specialized field of bio-medical sciences. This makes the use of tools like WordNet as is done in [1, 2] less effective. However contrary to general language queries, we have the advantage that the biomedical field is covered excellently by several ontologies. Combining these multiple semantic ontologies in a very simple heuristic to obtain an optimal query expansion, makes our approach distinct from previous approaches.

## 3 The solution

We have opted to address the issues raised above with some simple heuristics, which rely heavily on the fact that semantic platforms like DISCOVER bring together multiple data sources. While each data source separately does not have adequate power to discriminate, together they do.

We use the following algorithm to prune the synonym list

- **Retrieve documents** We retrieve all the documents that match the query string exactly either with their preferred label or with one of the alternative names. The provenance of each of these labels or names is also retrieved
- **Merge documents** the documents are merged if there is sufficient overlap between their names and if the classes of the documents are compatible. As an example consider a broad term like *lung cancer*. Multiple disease instances match this term. By merging these instances and their synonyms we cover the landscape the user probably wants to investigate
- **Score synonyms** Synonyms are scored based on the number of data set that support them

$$\text{score} = \frac{\#data\ sets - 1}{\max(data\ sets) - 1} \quad \text{if } \#data\ sets > 1$$

If *data sets* == 1 then the score is set 1. This scales the score between 0 and 1.

If a synonym is sufficiently short (currently 8 characters) then it is checked whether it gives rise to false positives. This happens by retrieving the classes of the documents having exact matches for that synonym. If more classes are found than the current concept possesses, it gets a negative score for being ambiguous. If a synonym is a number with less than 5 digits or less than 3 alphanumeric characters, it also gets a negative score for the same reason.

- **Remove containing synonyms** If a synonym contains another shorter synonym, there is no reason to put it in a query. If the shorter word has a lower score than the containing word, it inherits the highest score

We only retain the synonyms which have a score larger than 0.

In the example of aspirin we retain 12 possible synonyms (16%), the first three being *aspirin*, *acetylsalicylic acid* and *salicylic acid acetate*.

## 4 Evaluation and results

We have analyzed the results of the 500 most prevalent keywords which were recorded in DISCOVER in the first 6 months of 2017. For each keyword we have ordered all the synonyms by score and by descending word length in case of a tie for score. As an evaluation we submitted all these keywords to the search engine in this order instead of submitting only the optimal synonyms and we recorded the following per synonym

- How much hits we found in total by accumulating the synonyms with *OR*
- How much hits we get by only submitting the current synonym

An example of this output is shown in Appendix A. Per merged document we register

- The optimal number of synonyms. We do this by re-ordering the synonyms by descending number of hits they add and count how much synonyms would we minimally need to obtain 98% of the hits
- Total number of synonyms per merged document
- The number of synonyms with score  $> 0$

This enables us to measure what we miss if we only submit synonyms with a score  $\geq 0$ .

We can split the results in two big groups, which we analyze separately. On one hand we have keywords for which the score is not able to discriminate: so all synonyms need to be checked. Basically this happens when there is only one data source contributing to the concept or that all the data sources completely agree on all alternative names. On the other hand we have the group for which the method does make some difference and it does allow us to skip some synonyms.

The first group on average has 6.16 synonyms per keywords, while the second group has 30.60 synonyms per keyword. In other words, in the cases the method does not make any difference, there was not really a need for it to begin with.

Group	Average #synonyms	% submitted	Average missed hits	Median missed hits
All synonyms considered	6.16	100%	0	0
Filtered by method	30.6	15.8%	9.4%	1.7%

In the second group we submit only 15.8% of all the synonyms if we apply the method. We observe that the time to run the factor roughly scales with the amount of synonyms we submit. If we compare the number of hits we obtain by this subset to the hits by submitting all the synonyms, on average we miss 9.4%. However the median of the fraction of missed hits is 1.7%. So we have a few very high outliers. What is causing this? The worst is example is *DNM2* where we miss 94% of all the hits by only considering the synonyms with score  $> 0$ . This is almost exclusively caused by one alternative name for this gene i.e. *Cytoskeletal protein* (through UMLS). Although this gene is indeed related to this concept, this concept is much broader. So it is a *hypernymy* of the submitted concept and excluding it is actually beneficial: it would have given 16 times more false positives if it were included. This is a pattern for most high miss fraction examples. Consider *atezolizumab*. Here the hypernymy *anti-pd-l1* (from ChEMBL) is successfully excluded by our method. So we consider it justified to exclude the high end misses tail and focus on the median: more than 98% of the hits were found by less than 16% of the synonyms. The minimum number of synonyms needed to get 98% of the hits is actually 9,01%, meaning that we submit less than double of this absolute minimum.

For ambiguous synonyms we conducted a manual check for false positives in a small subset of 20 clinical studies prominently containing *ASA*. Nine out of twenty are not about aspirin at all: we found 3 about 5-aminosalicylates, 3 about the ASA-PS classification and 1 about resp. advanced surface ablation, Avonex- Steroid Azathioprine and Argininosuccinic Aciduria. Of the other 11 only one does not contain one of the other synonyms included by our method. So we avoid 45% false positives and trade these for 9% false negatives.

## 5 Conclusion

We have reduced the number of submitted synonyms by 84%, thereby losing only 1.7% of the hits. The false positive exclusion, which is a lot harder to check, also seems to work well based on the small manually curated sample.

On a broader level we see that the actual number of synonyms needed to attain most of the text hits is even lower: on average we only between 2 and 3 synonyms, the median is even 1! So for public data set it might be a hint to focus more on quality then on quantity when choosing alternative names for concepts.

Overall we can conclude that the methods works well, despite the fact that it is very simple. It is clearly a demonstration of the cheap gains we get by combining multiple data sets in one semantic framework.

## A APPENDIX Complete example: *Aspirin*

The example output we generated when evaluating *Aspirin* is presented here. The synonyms are submitted in order of the table 1, which also contains the results for each synonym. As you can see the first 2 synonyms bring the bulk of the hits. The second one (*Acetylsalicylic acid*) has about 8 times less hits than *Aspirin*. But of those 10000 hits only about third is unique: the other already have a hit for *Aspirin*. This pattern returns although all subsequent synonyms return even less hits. The only synonym returning a significant number of synonyms is *ASA*, but as mentioned in the text this is a very ambiguous word with many different meanings. Many of these hits are identified as false positives.

Overall *Aspirin* has 75 alternative names. In the table we omit the containing synonyms (such as *aspirin sodium*). In total 17.3% of the synonyms are submitted (13) and we miss 1.04% of the hits. In the optimal case only 2.7% of the synonyms have to be submitted to obtain 98% of the hits. The timings for the queries are: 70 ms for retrieving hits for *Aspirin* only, 190 ms for retrieving all hits for synonyms with score  $\geq 0$  and 2650 ms for all 75 synonyms.

synonym	(1)	(2)	(3)	(4)	(5)
aspirin	1	100.00%	100.00%	84527	84526
acetylsalicylic acid	0.71	4.08%	12.19%	87974	10308
salicylic acid acetate	0.43	0.00%	0.01%	87977	13
2-(acetyloxy)benzoic acid	0.43	0.01%	0.24%	87989	208
acetylsalicylate	0.29	0.25%	0.81%	88210	709
2-acetoxybenzoic acid	0.29	0.01%	0.02%	88218	21
o-acetylsalicylic acid	0.29	0.00%	0.02%	88218	19
o-acetoxybenzoic acid	0.14	0.00%	0.00%	88218	2
2-acetyloxybenzoic acid	0.14	0.00%	0.01%	88221	8
o-carboxyphenyl acetate	0.14	0.00%	0.00%	88221	1
acidum acetylsalicylicum	0.14	0.00%	0.00%	88223	4
benzoic acid, 2-(acetyloxy)-	0.14	0.00%	0.02%	88227	19
2-acetoxybenzenecarboxylic acid	0.14	0.00%	0.00%	88227	1
levius	0	0.02%	0.04%	88245	31
platet	0	0.11%	0.13%	88344	111
asprin	0	0.08%	0.18%	88416	162
enprin	0	0.00%	0.00%	88416	1
asparin	0	0.00%	0.00%	88418	3
danamep	0	0.00%	0.00%	88418	1
paynocil	0	0.00%	0.01%	88419	12

**Table 1.** All synonyms is DISCOVER for *aspirin*. Columns (1) Score, (2) Percentage of extra hits compared to all previous synonyms combined with OR, (3) Percentage of hits for this synonym only, (4) Number of total cumulative hits (all previous synonyms combined with OR), (5) Number of hits for this synonym only

synonym	(1)	(2)	(3)	(4)	(5)
measurin	0	0.00%	0.22%	88422	191
aspirine	0	0.10%	0.16%	88514	141
postmi 75	0	0.00%	0.00%	88514	3
gencardia	0	0.00%	0.00%	88514	1
aspro clr	0	0.00%	0.00%	88514	1
equi-prin	0	0.00%	0.00%	88514	1
disprin cv	0	0.00%	0.00%	88514	2
alka rapid	0	0.00%	0.00%	88514	1
postmi 300	0	0.00%	0.00%	88514	2
polopiryna	0	0.00%	0.01%	88518	7
acetophen	0	0.00%	0.00%	88518	2
angettes 75	0	0.00%	0.00%	88518	1
nu-seals 75	0	0.00%	0.00%	88518	2
nu-seals 300	0	0.00%	0.00%	88518	1
nu-seals 600	0	0.00%	0.00%	88518	1
8-hour bayer	0	0.00%	0.00%	88518	1
micropirin ec	0	0.00%	0.00%	88518	1
disprin direct	0	0.00%	0.00%	88518	1
acetosalic acid	0	0.00%	0.00%	88518	1
acetylsalic acid	0	0.00%	0.00%	88520	4
anadin all night	0	0.00%	0.01%	88521	12
2-acetoxybenzoate	0	0.05%	0.08%	88563	67
acetyl salicylate	0	0.02%	0.10%	88585	92
acetylsalicylsure	0	0.00%	0.02%	88589	19
nu-seals cardio 75	0	0.00%	0.00%	88589	1
azetylsalizylsure	0	0.00%	0.00%	88589	1
azetylsalizylsaeure	0	0.00%	0.00%	88589	1
acetylsalicylsaeure	0	0.00%	0.00%	88589	1
acetylsalisylic acid	0	0.00%	0.01%	88592	9
bayer extra strength	0	0.00%	0.00%	88592	2
acetylsalicyclic acid	0	0.29%	0.53%	88850	472
acetyl salicyclic acid	0	0.47%	1.07%	89264	949
cido acetilsaliclico	0	0.03%	0.05%	89290	43
acetyl salicyclic acid	0	0.00%	0.02%	89294	14
2-acetoxy-benzoic acid	0	0.06%	0.08%	89348	69
acide actylsalicylique	0	0.00%	0.01%	89348	5
acetylsalicylicum acidum	0	0.00%	0.01%	89350	10
acide 2-(actyloxy)benzoque	0	0.00%	0.00%	89350	1
acide 2-(acetyloxy)benzoique	0	0.00%	0.00%	89350	1
(aspirin)2-acetoxy-benzoic acid	0	0.00%	0.00%	89350	2
2-(methoxycarbonyl)benzoic acid	0	0.00%	0.00%	89353	4
ecotrin		0.01%	0.25%	89359	227
asa		42.19%	48.64%	127053	43461

Table 2. The continuation of table 1

## References

1. Voorhees, Ellen M. Query expansion using lexical-semantic relations. Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval. Springer-Verlag New York, Inc., 1994.
2. Mandala, Rila, Takenobu Tokunaga, and Hozumi Tanaka. "Combining multiple evidence from different types of thesaurus for query expansion." Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1999.