Creative language explorations through a high-expressivity n-grams query language

Carlo Strapparava⋆, Lorenzo Gatti◦, Marco Guerini◦, and Oliviero Stock⋆
⋆FBK-irst, Povo, Italy
◦ Trento-Rise, Trento, Italy
strappa@fbk.eu, gattilorenz@trentorise.eu, marco.guerini@trentorise.eu, stock@fbk.eu

Abstract

In computation linguistics a combination of syntagmatic and paradigmatic features is often exploited. While the first aspects are typically managed by information present in large n-gram databases, domain and ontological aspects are more properly modeled by lexical ontologies such as WordNet and semantic similarity spaces. This interconnection is even stricter when we are dealing with creative language phenomena, such as metaphors, prototypical properties, puns generation, hyperbolae and other rhetorical phenomena. This paper describes a way to focus on and accomplish some of these tasks by exploiting *NgramQuery*, a generalized query language on Google N-gram database. The expressiveness of this query language is boosted by plugging semantic similarity acquired both from corpora (e.g. LSA) and from WordNet, also integrating operators for phonetics and sentiment analysis. The paper reports a number of examples of usage in some creative language tasks.

Keywords: N-grams, WordNet, Semantic Similarity, Creative Language.

1. Introduction

Many tasks in computation linguistics are properly dealt with exploiting a combination of syntagmatic and domain features. Syntagmatic aspects are often managed by information present in large n-gram databases, while domain and ontological aspects are more properly modeled by semantic similarity spaces (e.g. latent semantic space) and lexical ontologies such as WordNet (Fellbaum, 1999). This interconnection is even stricter when we are dealing with creative language phenomena, such as metaphors, prototypical properties, etc.

This paper describes a way to focus on and accomplish some of these tasks by exploiting an improved version of *NgramQuery*, a generalized query language on Google N-gram database (Aleksandrov and Strapparava, 2012).

The expressiveness of this query language is boosted by plugging semantic similarity acquired both from corpora (e.g. LSA) and from WordNet. It contains several different operators, which combined in a proper query can help users to extract n-grams having similarly close syntactic and semantic relational properties. The tool can be useful in a variety of tasks, ranging from specific lexicon extraction and lexical substitution task (McCarthy and Navigli, 2007) to the automatization of creative and figurative language processes (Strapparava et al., 2007; Stock et al., 2008; Veale, 2011) such as puns generation, metaphors, hyperbolae and other rhetorical phenomena.

2. NgramQuery Language

We used and refined the *NgramQuery* generalized query language (Aleksandrov and Strapparava, 2012). The starting point was the Google Web 1T 5-Grams database (Brants and Franz, 2006). This data set contains English word n-grams and their observed frequency counts. The length of the n-grams ranges from unigrams to five-grams. The n-gram counts were generated from approximately 1 trillion word tokens of text from publicly accessible Web pages. The present version of *NgramQuery* combines several knowledge sources, in particular:

- lexical concepts and their taxonomies, given by *WordNet* lexicon database. Besides all the relations present in WordNet, we embedded in the query language also the common similarity measures such as Resnik, Lin, Jiang-Conrath, etc. (Budanitsky and Hirst, 2006; Pedersen et al., 2004).
- Similarity from a specific version of *LSA* space (Deerwester et al., 1990) acquired from the full British National Corpus.
- The *CMU Pronouncing Dictionary*[1], for dealing with assonances, partial homophones, etc.
- Valence scores of the synsets using SentiWordNet (Esuli and Sebastiani, 2006), and SentiWords' prior polarities (Guerini et al., 2013) for non-disambiguated terms. This is a newly introduced feature.

Finally we designed a query language that is able to express these concepts according to the Google N-gram database. Information extraction from such a database using fixed external resources is not straightforward, especially with respect to efficiency and completeness. To give a flavor of the expressiveness, a possible query could be:

$$Italy\#L\#20 \;\; spaghetti \;\; with \;\; \sim\#food\#n$$

that retrieves the 4-grams, along with their frequencies, in which the first word is one of the first 20 most similar words to *Italy* according to LSA (i.e. $Italy\#L\#20$), then '`spaghetti with`' followed by is a hyponym of the noun $food$ (i.e. the term $\sim\#food\#n$). The most frequent 4-gram (with frequency 150) that satisfies the query is:

[1] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

```
150 italian spaghetti with meat
```

In Table 1 the operators implemented for each of the WordNet categories are listed. We also introduced a new WordNet operator (expressed with the . symbol) that, given a lemma, retrieves the other synset members. See (Aleksandrov and Strapparava, 2012) for a full description of the query syntax.

Operator	**n**	**v**	**a**	**r**
Antonym (!)	+	+	+	+
Hypernym (@)	+	+	-	-
Instance hypernym ($@i$)	+	-	-	-
Hyponym ($\sim$)	+	+	-	-
Instance hyponym ($\sim i$)	+	-	-	-
Member holonym ($\#m$)	+	-	-	-
Substance holonym ($\#s$)	+	-	-	-
Part holonym ($\#p$)	+	-	-	-
Member meronym ($\%m$)	+	-	-	-
Substance meronym ($\%s$)	+	-	-	-
Part meronym ($\%p$)	+	-	-	-
Attribute ($=$)	+	-	+	-
Derivationally related form ($+$)	+	+	-	-
Entailment ($*$)	-	+	-	-
Cause ($>$)	-	+	-	-
Also see (^)	-	+	+	-
Verb group ($)	-	+	-	-
Similar to (&)	-	-	+	-
Participle of verb ($<$)	-	-	+	-
Pertainym (\)	-	-	+	-
Derived from adjective (\)	-	-	-	+
Member of the same synset (.)	+	+	+	+

Table 1: WordNet pointer-concept operators. Abbreviation are as follows: **n** (*noun*); **v** (*verb*); **a** (*adjective*); **r** (*adverb*); $+$ means the relation is present in WN, and $-$ otherwise.

We have considered that a typical use of the query language would be in two phases. A first phase is exploratory. It is concerned with human exploration of queries and their results. Typically one has certain ideas and expectations that need to be verified. The system offers the possibility of composing quite sophisticated requests; results as well may be surprising and possibly require reformulation of the query. For this phase, a graphical interface that allows easy formulation of the query, possible reformulation and exam of results is a very important resource. We shall briefly describe it here below. The second phase is a programmatic use of the query language. For that purpose the interface is not relevant, but the characteristics of the queries should have been well explored beforehand. Normally, in a program, the specific values of the query parameters are computed dynamically. In more complex programs, even the whole query expression may be dynamically built at execution time.

2.1. The graphical interface

We developed a graphical interface (Figure 1) to aid the explorative phase, so users can abstract from the precise syntax of the query language and focus on the high-level extraction mechanisms. Pointers, relations and other options are chosen from comboboxes, while similarity and SentiWordNet numerical values are selected with sliders. This way, even non-expert users can use *NgramQuery* for understanding how to properly formulate the query, and explore slight variations, while at the same time blocking malformed or impossible queries. Other useful features are provided too, like a history file where previous queries can be easily retrieved and cached results inspected, or the possibility to copy the query string in memory.

3. Creative Language tasks

This section shows some examples on how to use the query language and how it is possible to combine different resources within a query.

3.1. Single Query Examples

Characteristics of an entity. The expressivity of the query language allows us – for instance – to extract the typical characteristics of an object. Let us suppose that we want to find an attribute of Paris that has a positive valence (note the operator S for specifying sentiment scores):

$$Paris\ is\ a\ \#a\#S\#0.65;1.0\ city$$

that retrieves the 5-grams, along with their frequencies, in which as an attribute of *city* there is an adjective with a valence falling in the interval [0.65,1.0]:

```
347  Paris is a beautiful city
261  Paris is a wonderful city
165  Paris is a great city
```

If we ask for a more neutral adjective, e.g. in the range [0.0,0.5], we would get:

```
189  Paris is a multicultural city
152  Paris is a captivating city
```

Since we do not know the WordNet sense number of `beautiful`, `multicultural` and the other adjectives we are extracting with these queries, our valence score cannot be directly found in SentiWordNet, which is labelled at synset level. In this case, the script uses the SentiWords[2] prior polarities list (i.e. if that word out of context evokes something positive or something negative).

Prototypical objects. We can extract the objects that typically hold a specific property. For example, for getting those fruits considered sweet:

$$as\ sweet\ as\ a\ \sim\#fruit\#n$$

that retrieves the 5-grams, along with their frequencies, that represent the sweet 'hyponyms' ($\sim$) of the noun *fruit*.

```
141  as sweet as a mango
```

[2] http://hlt.fbk.eu/technologies/sentiwords

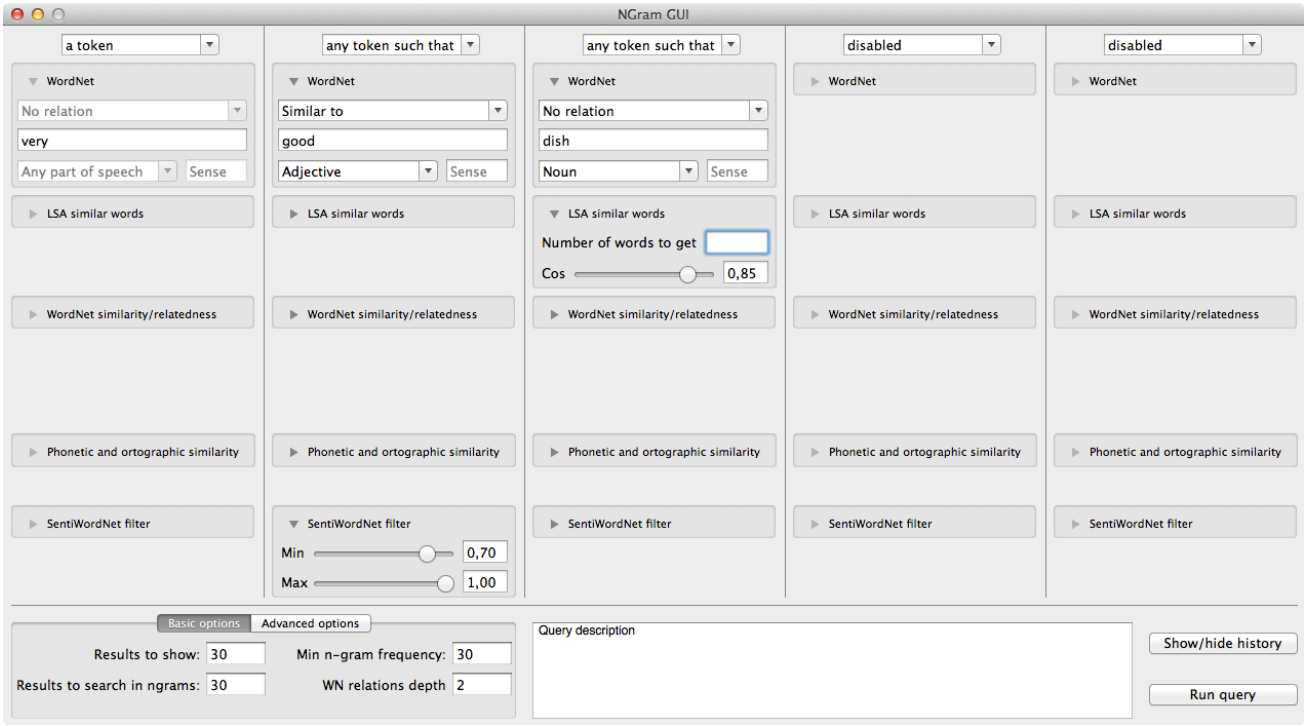

Figure 1: The *NgramQuery* GUI

Terminology extraction. Combining information in the n-grams and the hierarchy of WordNet, we can extract specific terminology and phrases related to a particular context.

$$\sim \#food\#n\#2 \;\; sandwich, pizza, pasta, salad\#n$$

that retrieves the 2-grams, along with their frequencies, that are composed of a hyponym of the second sense of the noun *food*, followed by any synonym of the nouns *sandwich*, *pizza*, *pasta*, or *salad*. We get the following results:

```
191865  pasta salad
 25152  fish sandwich
 20442  seafood pasta
 10146  fish salad
  8900  chocolate sandwich
  6881  meat pasta
  6277  meat sandwich
  5347  seafood pizza
```

Lexical substitution. In lexical substitution tasks we can use the 'member of the same synset' operator (.) to find substitutes for a given word. For example, if we want to change the adjective in *delectable food*, the query

$$.\#delectable\#a \;\; food\#n$$

retrieves the 2-grams of a synonym of the adjective *delectable*, followed by the noun *food*, e.g.

```
170467  delicious food
 31027  yummy food
  4292  delectable food
  3999  scrumptious food
```

Since we have the frequency of each expression, we can easily choose the most frequent (i.e., *delicious food*) or rather select something less used, depending on our purposes. If we want to relax the requirement of strict synonymy, we can exploit the Wordnet adjective operator 'similar to' (&):

$$\&\#yummy\#a \;\; food\#n$$

thus obtaining

```
47112  tasty food
```

3.2. Query Blending

Even if many tasks can be accomplished with a single query, blending the results of different queries can further expand the possibilities of the tool. For example, one could find differences (or similarities) in gender prejudices by comparing the results of the query "$men \;\; love \;\; \#n$" and "$women \;\; love \;\; \#n$" (i.e., the operator $\#n$ acts as a wildcard among the nouns, so finding names of things that are considered loved by men or women). The top results of the two queries in isolation, ordered by frequency, are shown in Table 2.

To extract gender specific prejudices, we need to take into account also the opposite gender results in order to rule out those objects that are "loved" by both males and females. To do so, we use a mutual information approach.

$$MI = \frac{f(object_x, propriety_y)}{\sum_i f(object_i, propriety_y)}$$

For each term we divide its frequency when associated to males by the frequency when associated to females or

men love #n	Frequency	*women love #n*	Frequency
bitches	5565	it	3059
war	2740	sex	2137
it	1949	men	2020
women	1865	advice	1460
football	849	latino	1329
jesus	791	me	1317
making	705	free	1168
darkness	697	cats	1089
advice	587	line	1059
tips	582	relationships	1017

Table 2: Information extraction by gender

males, and vice-versa. To account for terms that are not present in both cases, a Laplace smoothing with $\alpha = 20$ is used[3]. This mutual information approach is of particular use when multiple objects can be compared, for example to extract the prototypical characteristics of cities. If we consider only n-grams frequencies, all important cities tend to be *expensive*, while considering MI, we can discover that Rome is *eternal*, while Cambridge is *posh*, London is *expensive* and if you want some *romance* you should go to Venice.

3.3. Metonymy generation

Metonymy is a figure of speech in which one word is substituted for another, closely associated in meaning. Let us focus, for example, on the task of generating one particular type of metonymy, i.e. the Artist-for-Artform (Fass, 1991). To use it in the sentence *the orchestra played a beautiful sonata*, we could use a query such as:

$$played\ \ sonata\#n\#L\#20$$

which retrieves the 2-grams with *played* and the 20 words most LSA-similar to *sonata*:

```
30787  played piano
 8575  played violin
 2704  played brahms
 1867  played mozart
 1675  played beethoven
 1584  played bach
 1086  played chopin
```

The output can be filtered in many ways. One possibility is again query blending, by exploiting WordNet's hierarchy as in (Harabagiu, 1998), and checking that, if the object we want to change is a hyponim of "classical music" (i.e. $i\#classical\_music\#n$) output has to be an instance hyponym of composer ($\sim i\#composer\#n$). This of course requires some specific knowledge that maps music to composers, paintings to painters, and so on. Another possibility is given by Named Entity Recognition modules or gazetteers.

Whatever the adopted solution is, the final list will contain only *mozart*, *beethoven*, *bach*, *chopin*, while *piano* and *violin* will be discarded.

[3]Since in the Google Web 1T 5-Grams the n-grams frequency cutoff is 40, we use the mean of possible values as an estimation.

4. Further work

We are actively developing new features for *NgramQuery*, the main one being an option for reordering the output according to different measures, instead of simply considering n-gram frequency. For example, if we need to find a positive adjective for *man*, we could ask for all the positive adjectives that precede it. However, this means that, for most of the words, we will choose *good* as the positive adjective, since it is very generic and very common. To choose a more specialized word we can use, for example, the Pointwise Mutual Information or other mutual information measures (Bouma, 2009) that take into account also the frequency of the single words, and possibly rank *righteous* higher than '*good*'.

Another interesting feature is the ability to specify a set of words by extension (writing all the members of the set, e.g. {*cat, dog, hamster*}) instead of relying on the WordNet hierarchy, where no relation connects the set of three animals of this example. This allows us to easily extract things with common-sense knowledge by defining new sets on-the-fly.

The system can be a fundamental resource within complex applications. For instance one case is a creative system which is meant to promote a product or a concept by yielding a new positive attribute every day. The attribute could bring with it an analogy with the best fitting city, as in the example described in section 3.2 so that the system comes out everyday with expressions such as: "This X is as Y_i as Z_i" where Y_i is an adjective which expresses a distinguished positive characteristic of city Z_i. Y_i can be found automatically as shown above, and the system iterates on the list of cities, so that every day it comes out with a new one. For instance "this car is as romantic as Venice", "this car is as frenetic as New York".

Another example of a potential application is in producing an assessment of popular sentiment about X. For instance this could help compiling a list of prejudices in our society, e.g. "many people think X are Y", with Y constrained to be a negative attribute. It could then possibly produce expressions that could counter automatically those prejudices; for instance "Many X are Z. By the way are you sure you are not Y?" with Z a positive attribute found for X, and Y the one above.

Finally, it is worth noting that the current version of *NgramQuery* is used within *Valentino*, a tool for the affective variation of existing texts (Gatti et al., 2014).

5. Conclusions

Thanks to various resources now available, we can combine within one single query language the possibility of retrieving the most popular expressions that have been actually used, in agreement with constraints that come from reasoning on lexical knowledge. The combination of various different constraints provides an enormous potential capability to find popular solutions of natural language use. For an advanced exploitation of the query language for specific applications, two phases are typically needed: a) an exploratory phase for understanding the results and tuning up the query, and b) the insertion of the right query (and exploitation of the results) in the final application program, with parameters typically defined at run time. We have also

developed a graphical tool that facilitates the exploration phase so that we can get to the final applied use quickly and creatively.

Acknowledgement

This work was partially supported by the PerTe project (Trento RISE).

6. References

Aleksandrov, M. and Strapparava, C. (2012). Ngramquery - smart information extraction from google n-gram using external resources. In *Proceedings of the 8^{th} international conference on Language Resources and Evaluation (LREC-2012)*, Istanbul, Turkey, May.

Bouma, G. (2009). Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL*, pages 31–40.

Brants, T. and Franz, A. (2006). Web 1T 5-gram version 1. Linguistic Data Consortium.

Budanitsky, A. and Hirst, G. (2006). Evaluating WordNet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32:13–47.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407.

Esuli, A. and Sebastiani, F. (2006). Sentiwordnet: A publicly available lexical resource for opinion mining. In *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06)*, pages 417–422.

Fass, D. (1991). Met*: A method for discriminating metonymy and metaphor by computer. *Comput. Linguist.*, 17(1):49–90, March.

Fellbaum, C. (1999). *WordNet*. Wiley Online Library.

Gatti, L., Guerini, M., Stock, O., and Strapparava, C. (2014). Sentiment variations in text for persuasion technology. In *Proceedings of the 9th International Conference on Persuasive Technology*.

Guerini, M., Gatti, L., and Turchi, M. (2013). Sentiment analysis: How to derive prior polarities from SentiWordNet. In *Proceedings of EMNLP 2013*.

Harabagiu, S. (1998). Deriving metonymic coercions from wordnet. In *Workshop on the Usage of WordNet in Natural Language Processing Systems*, pages 142–148.

McCarthy, D. and Navigli, R. (2007). The semeval English lexical substitution task. In *Proceedings of the ACL Semeval workshop*.

Pedersen, T., Patwardhan, S., and Michelizzi, J. (2004). WordNet::Similarity - measuring the relatedness of concepts. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI-2004)*, pages 1024–1025, San Jose, CA, July. Lawrence Erlbaum Associates.

Stock, O., Strapparava, C., and Valitutti, A. (2008). Ironic expressions and moving words. *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, 22:1045–1057.

Strapparava, C., Valitutti, A., and Stock, O. (2007). Affective text variation and animation for dynamic advertisement. In *Proccedings of 2^{nd} International Conference on Affective Computing and Intelligent Interaction (ACII2007)*, Lisbon, Portugal, September.

Veale, T. (2011). Creative language retrieval: A robust hybrid of information retrieval and linguistic creativity. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 278–287, Portland, Oregon, USA, June. Association for Computational Linguistics.