Automatic generation of semantic corpora for improving intent estimation of taxonomy-driven search engines

Lorenzo Massai*

Università degli Studi di Firenze, Department of Mathematics and Computer Science, Firenze, Italy

ARTICLE INFO

Keywords:
Natural language processing
semantic expansion
automatic query reformulation
pseudo-relevance feedback
search result clustering
indexing
search engines
intent estimation
semantic clustering

ABSTRACT

With the increasing demand of intelligent systems capable of operating in different user contexts (e.g. users on the move) the correct interpretation of the user-need by such systems has become crucial to give a consistent answer to the user query. The most effective techniques which are used to address such task are in the fields of natural language processing and semantic expansion of terms. Such systems are aimed at estimating the actual meaning of input queries, addressing the concepts of the words which are expressed within the user questions. The aim of this paper is to demonstrate which semantic relation impacts the most in semantic expansion-based retrieval systems and to identify the best tradeoff between accuracy and noise introduction when combining such relations. The evaluations are made building a simple natural language processing system capable of querying any taxonomy-driven domain, making use of the combination of different semantic expansions as knowledge resources. The proposed evaluation employs a wide and varied taxonomy as a use-case, exploiting its labels as basis for the expansions. To build the knowledge resources several corpora have been produced and integrated as gazetteers into the NLP infrastructure with the purpose of estimating the pseudo-queries corresponding to the taxonomy labels, considered as the possible intents.

1. Introduction

Information retrieval architectures often deal with the interpretation of the user query, especially when it comes to queries which can be posed in natural language. The main holdback is the identification of a connection between unstructured questions and structured data, which is an open problem and main task of natural language processing applications. The estimation of the user intent in natural language queries can be a difficult task and it can't be achieved considering only the syntactic representation of terms; a more semantic characterization is needed to express the *sense* and *context* of terms. The applications to get such characterizations are mainly in the fields of machine learning and semantic expansion [Navigli and Ponzetto (2012)], [Fellbaum (1998)], addressing problems like context learning, term disambiguation and polysemy. Semantic characterization of terms can be achieved, among the others, through data aggregation exploiting automatic clustering techniques [Lloyd (1982)], [Cortes and Vapnik (1995)] or ontologies [Bhogal et al. (2007)], which provide a qualitative representation of documents through semantic data models. The different strategies are analyzed in Section 2. The aim of this paper is to provide an automatic procedure to generate semantic resources supporting document retrieval in any domain and evaluate a generic model using them; such procedure is described in Section 3. Different strategies for building the knowledge resources are evaluated in Section 4, demonstrating both the most impacting semantic relation in document retrieval and the most effective combination of them. The resources have been generated by the means of each semantic relation available through state-of-the-art semantic repositories and the assessment highlights the implications of using each expansion when building a system based on term expansion. The evaluation has been assessed loading the generated corpora into a GATE pipeline [Cunningham et al. (2002)] to show the best tradeoff between the improvement and the worsening of the retrieval capabilities using different semantic resources. The data resources which have been exploited for the evaluation have been found in the range of Points of Interest (POIs) search engines.

2. Related work

In the field of query reformulation and features augmentation a key role is held by semantic networks and corpora repositories. The relations between terms can be used to get the context and sense behind a word in a phrase and the analysis of word frequencies in wide knowledge bases can be exploited as well to get latent semantics. The target of query expansion is giving relevance not only to the syntactic form of the words which are present within the query, but also to the most representative meaning of the words expressing the user-need. The most recent strategies for query augmentation [Azad and Deepak (2019)] include automatic and user-supported expansions and the main techniques adopted to perform an effective augmentation are interactive query expansion, pseudo-relevance feedback, search results clustering and semantic networks. Query expansion typically relies both on semantic networks and knowledge bases to get the relations between concepts.

2.1. Query expansion techniques

Query expansion is a field of natural language processing which aim is to provide an augmentation of the input terms composed by their most related concepts. A query expansion tool can be exploited to improve the retrieval ca-

*Corresponding author
lorenzo.massai@unifi.it (Lorenzo Massai)
ORCID(s): 0000-0002-8252-0549 (Lorenzo Massai)

pabilities of a search engine. There are many approaches to get such augmentation which can be classified as global analysis and local analysis [Lopes and Gadge (2014)]. The global methods are independent from the query and include interactive and automatic expansions. Local methods use the original query to retrieve documents and the query terms are augmented basing on a relevance feedback made by the users on the result-set or estimated. Semantic networks can be exploited to get closer to the semantics of the query terms considering the senses of each term basing on its classification. A similar effect can be achieved through search result clustering, which can be performed to estimate the classes (concepts) considering the semantic relatedness between terms as distances between them. Also, latent semantic analysis techniques [Deerwester et al. (1990)] can be used to expand the feature space estimating the concepts which are related to terms by the relations among them.

Interactive query expansion is a query expansion technique which consists in expanding the original query and getting a relevance feedback from the user on the relevance of a retrieved set of documents [Efthimiadis (2000)]. The set of retrieved documents is evaluated by the user, who expresses a choice selecting the best match for his/her need and explicitly refines the query formulation (Figure 1). Such method is effective to meet the user-need, but it is depending on user interaction and context which are, in general, prone to ambiguity.

Pseudo-relevance feedback is a refinement of relevance feedback, adding an automated step which avoids the user contribution in the process of query expansion [Baeza-Yates and Ribeiro-Neto (1999)]. The data flow from the user query to the retrieved documents includes the variation of the weights of query terms assuming that the top retrieved documents are relevant and using the position of the documents as a feedback for relevance (Figure 2). Pseudo-relevance feedback relies on the top documents retrieved by the logic itself, hence the method is heavily dependent on the ranking algorithm.

An application of relevance feedback is the Rocchio algorithm [Rocchio (1971)], a nearest centroid technique for classification that assigns to documents the class label of the training samples whose mean is closest to the document's class, modifying the weights of the vectors representing both the query and the documents. Rocchio classification can sometimes reflect the importance of a document respect to the whole collection rather than to the user query, which is potentially an unwanted behavior.

Search result clustering consists in partitioning the set of documents to retrieve in similarity classes and using them as an index for the actual retrieval. Text clustering aims at grouping the documents to retrieve in semantically related sets and is usually performed online. Such task can be assessed using a data-centric approach (data agglomeration techniques like K-means [Lloyd (1982)] and SVM classifiers [Cortes and Vapnik (1995)] which are depicted in Figure 3), or a description-centric approach (semantic agglomeration like ontology classes). The aggregation obtained through the clustering process can be considered as additional information expressing the semantics of the documents which are in the same cluster, therefore a feature for the original query terms. Such strategy is presented in [Kurland et al. (2006)], in which the cluster labels are used as an expansion of the original query terms. It is to be noticed that the expansion of query terms can potentially result into a deviance from the user-need which is known as query drift [Mitra et al. (1998)], because automatic expansion adds to the user query the most representative and semantically close meanings of such terms dealing with polysemy, term disambiguation and user context [Navigli and Ponzetto (2012)], [Bai et al. (2005)]. To avoid such deviation, the tradeoff between improvement and worsening of a retrieval system based on different query expansions has to be evaluated.

2.2. Semantic networks and knowledge bases

A **semantic network**, or **frame network** is an abstract representation of the semantic relations between concepts. The usual representation of the network is a directed or undirected graph containing nodes as concepts and arcs as semantic relations (Figure 4). Such representation aims at expressing the semantics of the terms which are associated to the concepts, since the relations among concepts are held for the associated terms. The conceptualization of a term is strictly related to the context in which it is used, and which could be unstated by the terms alone. The separation of different concepts is expressed by semantic networks having different and separated elements for those concepts which do not share the same semantic relations. As an example, in a semantic network describing the biology context the term *tree* will occur having the relation of hypernymy linked to the term *oak*; in the context of algebra the same term *tree*,

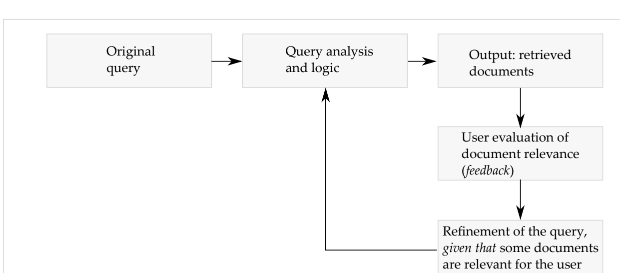

Figure 1: Relevance feedback

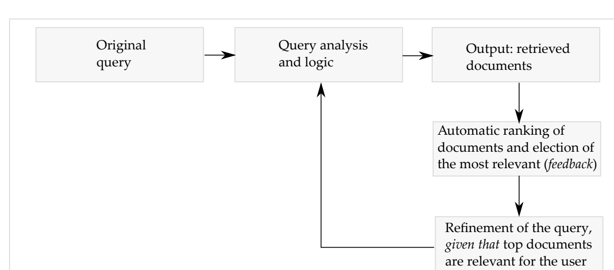

Figure 2: Pseudo-relevance feedback

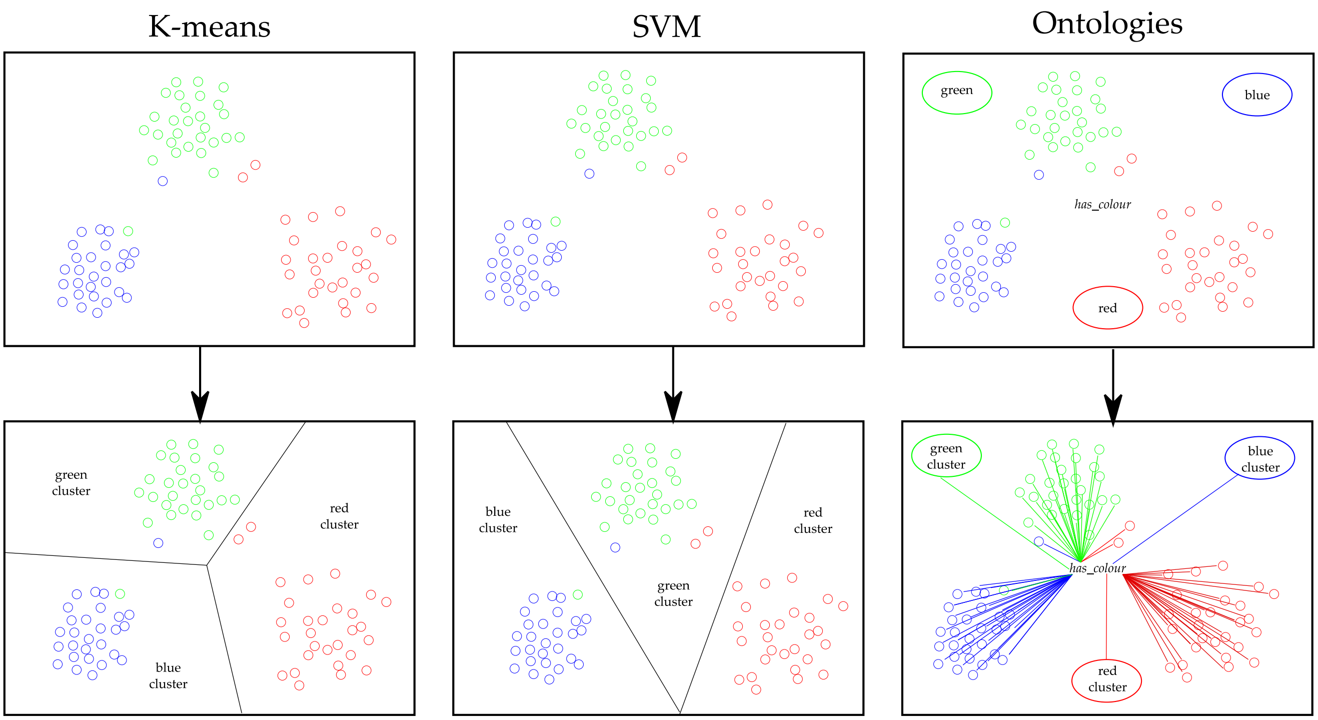

Figure 3: Clustering techniques and semantic classification

which expresses an *algebraic structure*, will not be associated to the term *oak*. The different semantic characterization expresses the difference between the two concepts of *tree*.

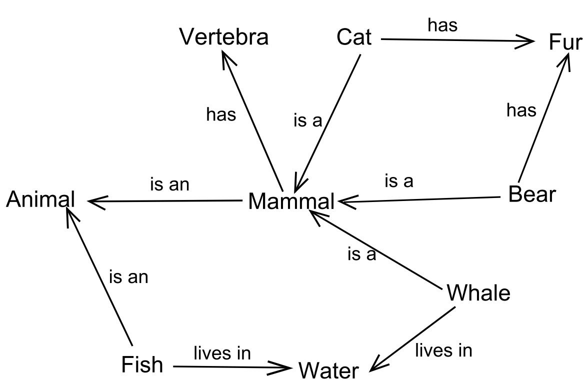

Figure 4: Semantic network for concept *mammal*

A **knowledge base** is an instantiation of data in a structured representation of concepts (A T-box and an A-box of an ontology form a knowledge base); such representation is organized by the relations between the concepts of the data which it can host. The data that is stored in a knowledge base can be retrieved by its properties; this kind of structure is also referred as a populated ontology.

2.2.1. WordNet

The most representative semantic network is WordNet [Fellbaum (1998)], a lexical database for English. It groups English nouns, verbs, adjectives and adverbs into synsets, each expressing a distinct concept. Synsets are interlinked by the means of conceptual-semantic and lexical relations. WordNet is similar to a thesaurus, since it groups words together basing on their meanings. However, there are some important distinctions. First, WordNet interlinks not just lexical form of words, but specific senses of words. As a result, words that are found to be near one to another in the network are semantically disambiguated. Second, WordNet labels the semantic relations among words, whereas the groupings of words in a thesaurus does not follow any explicit pattern other than meaning similarity. Some of the most representative semantic relations which are covered are *holonymy*, *hyponymy* (or *troponymy*), *synonymy*, *antonymy*, *hypernymy* and *meronymy*. WordNet web endpoint is available at: https://wordnet.princeton.edu/.

- **Synonymy** is the relation which includes the terms which are different from the input word, having the same meaning. The synonyms are obtained retrieving the terms contained within the same WordNet synset (e.g. computer, *desktop*).
- **Antonymy** is the relation which includes all the terms with opposite meaning with respect to the input word (e.g. black, *white*)
- **Hypernymy** is the relation which includes all the terms which constitute the more general category respect to the input word (e.g. dog, *animal*)
- **Hyponymy** is the relation which includes the terms with a similar, but less specific meaning (e.g. cd, *disc*)
- **Meronymy** is the relation which does not consider the

Table 1
WordNet semantic relations

Relation ID	Relation name	Relation meaning	Example
SYN	synonymy	*A* ***is the same as*** *B*	*ocean* → *sea*
ANT	antonymy	*A* ***is the opposite of*** *B*	*late* → *early*
SPC	hypernymy	*A* ***is kind of*** *B*	*gondola* → *boat*
GEN	hyponymy	*A* ***is more general than*** *B*	*boat* → *gondola*
COM	holonomy	*A* ***comprises*** *B*	*car* → *accelerator*
PAR	meronymy	*A* ***is part of*** *B*	*trunk* → *tree*

input as its specific meaning, but of the more general concept it represents (e.g. head, *boss*)

- **Holonomy** is the opposite relation to meronymy and it not considers the input as its meaning, but the more specific concept it represents (e.g. boss, *head*)

Other semantic networks like Babelnet [Navigli and Ponzetto (2012)] feature query expansion and disambiguation through the generation of semantically related sets of words called synsets (cognitive synonyms). The Babelnet framework provides term augmentation and transformation of each query term into a synset gathering data from WordNet, Open Multilingual WordNet, OmegaWiki, Wiktionary, Wikidata and Wikipedia (Figure 5).

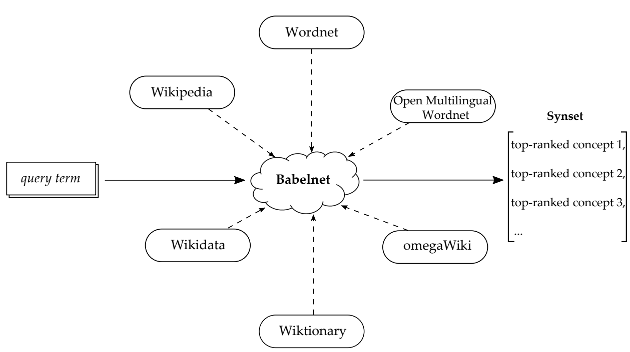

Figure 5: Resources exploited by the Babelnet semantic expansion

Each element of the synset is denoted as the entity which is most related to the concept of the term (i.e. its sense) within the Babelnet semantic graph. A concept is *most related* to another if it minimizes some distance (respect to each semantic relation) with any other contained in the networks outlined above.

2.2.2. Google Books N-grams dictionaries

Google Books is an endpoint for browsing the books data indexed by Google and it directly allows the retrieval of book texts. N-grams are fixed-size tuples of items, and in this case the items are words extracted from the Google Books corpus. The N specifies the number of elements in the tuple: a 5-gram contains five words or characters. The N-grams are produced by passing a sliding window on the text of books indexed by Google and giving as output a record for each new token. Such utility can be used for example to extract the terms that frequently surround an input word (e.g. *friend* → *best*). The Google Books N-grams data is indexed by several endpoints like Datamuse API [Datamuse (2016)] and it is accessible configuring the API with the proper code. This access point can be exploited to extract the popular nouns with input adjectives, the popular adjectives with input noun, the popular associated words, the frequent follower word and the frequent predecessor word.

- The **popular nouns with input adjectives** is a relation between the input words, which are adjectives, and the nouns which the adjectives are referred to (e.g. brown *eyes*, hot *water*);
- The **popular adjectives with input noun** is the same as the former, with nouns as input (e.g. *brown* eyes, *hot* water);
- The **popular associated words** is a relation which links to the input terms the words often related to them (e.g. bottle of *water*, glass of *wine*);
- The **frequent follower word** is a relation which bridges the input terms to words which are often following them (good *morning*, welcome *to*);
- The **frequent predecessor word** is the same as the former, linking the input to the words which are often preceded by the input terms (*good* morning, *welcome* to).

2.2.3. Onelook dictionaries

The Onelook Dictionary [Clink (2007)] search is a web endpoint and API which allows the user to look for definitions and related words to the input words or phrase. The search engine indexes more than 19 million words in more than 1000 online dictionaries, like Oxford Dictionaries, Merriam-Webster, American Heritage, Collins English Dictionary, Macmillan Dictionary, Wiktionary, Online Etymology Dictionary, Wikipedia, Rhymezone, WordNet 1.7, Free Dictionary, Dictionary.com, Wordnik and many others. The dictionaries can be browsed by topic: art, business, computing, medicine, miscellaneous, religion, science, slang, sports, technology and the definitions can be restricted to Chinese, English, French, German, Italian and Spanish. Onelook dictionaries endpoint is available at: https://www.onelook.com/.

Table 2
Google Books N-grams relations

Relation ID	Relation name	Relation meaning	Example
JJA	popular nouns	popular nouns modified by the given adjective	*gradual* → *increase*
JJB	popular adjectives	popular adjectives used to modify the given noun	*beach* → *sandy*
TRG	triggers	words that are statistically associated with the query word in the same piece of text	*cow* → *milking*
BGA	frequent followers	w' such that $P(w'\|w) \geq 0.001$	*wreak* → *havoc*
BGB	frequent predecessors	w' such that $P(w\|w') \geq 0.001$	*havoc* → *wreak*

2.3. NLP resources and tools

Keras is a high-level Python API based on neural networks that runs on CPU or GPU. It supports convolutional and recurrent networks and it can be used for text classification, text generation and summarization, tagging, parsing, machine translation, speech recognition, and other tasks. Keras runs on top of many frameworks and numerical computation libraries like **TensorFlow**, which is an open-source machine learning library by Google allowing GPU computation. The main concept behind TensorFlow is flow graphs usage: the nodes of the graph reflect mathematical operations, while the edges represent multidimensional inter-communicating data arrays (tensors). One of the most known of TensorFlow's NLP applications is Google Translate. Other applications are text classification and summarization, speech recognition, tagging, and so on. Considering the neural networks approach, **PyTorch** is a fast library which builds neural networks on a tape-based autograd system and provides tensor computation with strong GPU acceleration. Recurrent neural networks are mostly used in PyTorch for machine translation, classification, text generation, tagging, and other NLP tasks. The **Dynet** neural network library by Carnegie Mellon University adds the feature of handling syntactic parsing, which makes it an attractive choice for deeper analysis of the phrase. It supports C++, Python languages and GPU computing. Dynet is based on the dynamic declaration of network structure and over syntactic parsing it addresses also machine translation and morphological inflection analysis.
Stanford's **Core NLP** is a flexible and fast grammatical analysis tool that provides APIs for most common programming languages including Python and Java; it can also be run as a web service. The framework has features like part-of-speech (POS) tagging, named entity recognizing (NER), parsing, co-reference resolution, sentiment analysis, bootstrapped pattern learning, and open information extraction tools. Looking at feature separation and modularity, **GATE** is a Java framework for NLP which is organized as a pipeline: the elements of the pipeline can be added activating some features of the pipeline, which is run as a script. GATE provides lemmatization, POS-tagging, stemming, tokenization and many others, relying on low-level libraries which can be loaded in the pipeline, like the multilanguage TreeTagger POS-tagger, the Snowball stemmer, or the Stanford parser.
Looking at optimization and deep learning, **Deeplearning4j** is a Java programming library which can make use of the Keras API through TensorFlow. The main features of Deeplearning4j are the use of distributed CPUs and GPUs, parallel training via iterative reduce and micro-service architecture adaptation; moreover, vector space modeling enables solving of text-mining problems. It features POS-tagging, dependency parsing and word embedding through *word2vec*, a simple neural network which takes a corpus as input and computes vectors representing the semantic distribution of the words.

3. Generation of the document sets

The focus of this section is describing the procedures which have been employed to generate several sets of documents, each based on a specific semantic relation. The produced documents can be exploited to improve retrieval systems aimed at retrieving elements indexed by a taxonomy, like ontology-based systems or systems which are organized by hierarchical structures. In particular, the GATE framework and the Datamuse endpoint for accessing semantic repositories have been employed to reformulate a user natural language question into a category label of the target taxonomy. The GATE employment is twofold: in the resource generation phase (Section 3.2 and 3.3) and in the evaluation phase (Section 4) in which it is used to retrieve the most relevant documents respect to a user query.

3.1. Technologies and resources

To analyze benefits and drawbacks of employing a corpus built by the means of a specific semantic relation respect to others, the GATE framework for natural language analysis has been exploited. GATE is a solution flexible enough to allow the tuning of resources regardless of the NLP logic and it is capable of providing consistent analysis relying on external resources called gazetteers. The Datamuse API are used as an interface to generate such resources, querying semantic repositories such as WordNet, Google Books N-grams and Onelook dictionaries with respect to 11 different semantic relations.

3.1.1. Use-case data resources

Since the application domain for a use-case is a finite set of elements, a real taxonomy has been employed, under the assumption that the category and macro category labels of the taxonomy briefly describe the elements in the domain. Such labels are considered as all the possible intents of a user query within a fixed domain. The taxonomy which has been employed for the use case is the Yelp economical activities taxonomy, available as a json file in the Yelp website. The Yelp taxonomy well describes the commercial activities and services domain, expressing hierarchical relations between the service type (*parent*) and the service name (*title*) (e.g.: the *Food* service type is parent of several child categories with titles *Beverage store*, *Cake Shop*, etc.). The relation between the child categories and the parent macro categories is the *is-a* relation. Under the assumption described above, the association between a user natural language query and an actual set of documents described by a category label is shifted to the association of the user query to the correct label of the taxonomy (Section 4). The reformulation of a question as a category label can then be directly used as input for a data structure like an ontology to retrieve the actual data.

3.1.2. NLP technologies and utilities

The resource generation process is implemented using the GATE framework, exploiting its features to build a refinement and augmentation procedure to generate several corpora which contribution in document retrieval is assessed in the evaluation phase (Section 4). Each generated corpus is composed by a set of 1565 gazetteers generated from the Yelp taxonomy and containing category-related terms. The produced corpora maintain the *macro category/category* relation provided in the Yelp taxonomy as the *folder/file* relation. The categories are represented by the labels of commercial activities, services, public administrations, public transportation lines, etc. which are included within the taxonomy; such knowledge resources can be integrated in different forms, since they are not tied to the retrieval architecture.
To provide an evaluation of the impact of different kinds of expansion, the resources have been built in different ways by the means of all the available semantic relations. The endpoint which has been employed for accessing semantic databases such as WordNet, Google Books N-grams and Onelook dictionaries is the Datamuse API interface [Datamuse (2016)]. Such interface allows to specify as parameters one or more words, a *rel_[ID]* constraint representing the semantic relation to perform and an optional topic parameter to specify the domain of the word.

3.2. Generation of the evaluation corpus

To give evidence of the retrieval effects in using different semantic expansion criteria a set of lists has been generated, each containing as text (and named as) a category label. Each file is included in a folder named as its supercategory respect to the employed taxonomy, preserving the taxonomy hierarchy. This set of lists is considered as a bottom line for the evaluation, since it exploits not any semantic expansion, representing the same as pattern matching in a retrieval system based on gazetteers. To evaluate the impact of each semantic relation in user-intent estimation the bottom line has been expanded using the Datamuse API endpoint. The expansion has been performed by the means of WordNet *synonymy*, *antonymy*, *hypernymy*, *hyponymy*, *holonomy* and *meronymy* relations and the Google Books Ngrams *frequent words associations* (Table 1, Table 2). A fully automatic procedure to extract the related words by the means of each semantic relation has been produced (its architecture is shown in Figure 6) and the generated corpus for the input taxonomy is available for research purpose at: https://github.com/lmassai/semRels.

3.2.1. Software implementation

The algorithm is composed by four main phases which are: `generateBaseLine`, `generateListsBySemRel`, `enhanceLists` and `getLeavesToSort`.

1. The aim of the first phase is the generation of the bottomline, defined as a set of files corresponding to the taxonomy categories, organized by folders corresponding to the taxonomy macro categories. To build such resources consistently with the structure of the more complex ones the files have been filled with the category labels without any semantic expansion, filtered by not-relevant parts of speech like prepositions, conjunctions, articles and punctuation and trimmed by the words which are shorter than 2. The remaining words have been grouped and weighted by the number of occurrences. The aim of this phase is creating a corpus which makes the retrieval system behave as a pattern matching system once loaded in the GATE pipeline, serving as a bottom-line for evaluating the improvements of each expansion.
2. The `generateListsBySemRel` phase makes use of the Datamuse API to get the related words by the means of the relations described in Section 2, for each cluster label. Each label of the taxonomy is expanded exploiting the WordNet relations of *synonymy*, *antonymy*, *hypernymy*, *hyponymy*, *holonomy* and *meronymy* generating a corpus for each input semantic relation. Further expansions are provided querying the Onelook dictionaries and the Google Books Ngrams by the means of the *popular nouns with input adjective*, *popular adjectives with input noun*, *popular associated words*, *frequent follower word* and *frequent predecessor word* relations. These relations have been found to give a significant improvement to context estimation. The relations shown in Table 3 are also available through the Datamuse APIs and have not been included since they are not considered as an improvement for the conceptualization of a term.
3. The `enhanceLists` phase is the most complex, and it is aimed at shaping the content of the files generated in the previous phase to meet the requisites of gazetteer-based retrieval systems and at enhancing retrieval capabilities through data augmentation. Stop words such as prepositions, conjunctions, articles and punctuation are pruned through POS tagging and further filtering; the words contained in each list are provided with the original lemma, including the macro category labels and the category labels computed in the first phase. Finally, a weight is assigned to the non-label words (for which the weight is fixed) by assigning them a mea-

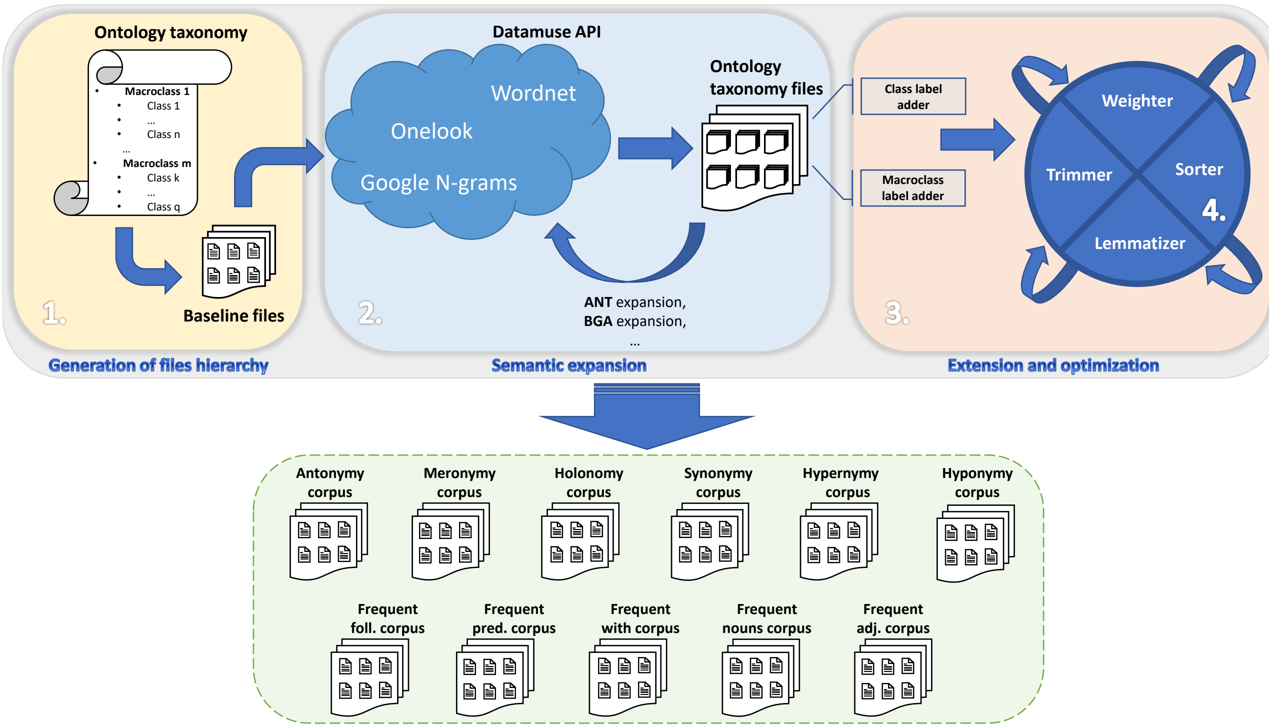

Figure 6: Generation of the corpus

sure of their semantic relatedness respect to the class label. Such measure is considered as the frequency of the term within the gazetteer after the expansion and the weights are assigned grouping identical words into one and increasing its weight for each trimmed word. The above described procedure is implemented by six sub-phases:

(a) the words related to the category labels (phase 2) are included;
(b) the name of the macro category is added;
(c) a part-of-speech filtering is obtained through the TreeTagger POS-tagger; [Schmid (1994)]
(d) the words which are shorter than 2 characters are removed;
(e) the lemmas of the terms are included;
(f) the duplicated words are grouped, and followed by the former number of occurrences as a weight.

As a result of this phase, 11 folders are produced, one for each semantic relation described in Section 2, each maintaining the original cluster structure in the form of folders and files which names correspond to the Yelp macro categories and categories labels. Each file contains its label and folder as text, followed by the enhanced output of the semantic repositories obtained through the Datamuse endpoint. It is to be noticed that some of the features included in this phase are also applied to the bottom-line files to provide coherence with the rest of the evaluation (e.g. removing the words which are shorter than 2 and making each grouped word followed by its number of occurrences).

4. The last phase aims at providing the most optimized performance in browsing the files, post-processing the lists and sorting them in alphabetical order to be retrieved through a binary search implementation.

The Datamuse API requests are formed using as parameters the *relation ID*, the *category label* of the Yelp taxonomy and the *macro category* to provide a more specific field for the expansion.

The procedure is highly scalable, and it can be easily applied to any real context or domain (e.g. using the T-box of an ontology as input taxonomy, or menu items as category labels). Loading the generated resources in a simple framework like the one which has been proposed in Section 4 for the evaluation allows to query any kind of data in natural language knowing only its structure, applying the same procedure to the words which are contained within the actual documents.

3.3. Combination of the generated resources

To provide evidence of the best performing combination of the analyzed semantic relations in intent estimation and to evaluate the best tradeoff between the highest accuracy and low deviance from the intent, the resources described in Section 3.2 have been combined in different ways. The combination criteria are all the possible combinations without repetitions, providing $\binom{n}{k}$ different configurations with $n = 11$, which is the number of the semantic relations that are present in the corpus generated in Section 3.2, and k which is the number of combining relations.
The first case for $k = 1$ is implemented by the result achieved

Table 3
Not-semantic relations

Relation ID	Relation name	Relation meaning	Example
RHY	rhymes	perfect rhymes	*spade* $\rightarrow$ *aid*
NRY	approximate rhymes	approximate rhymes	*forest* $\rightarrow$ *chorus*
HOM	homophones	sound-alike words	*course* $\rightarrow$ *coarse*
CNS	consonant match	consonant match	*sample* $\rightarrow$ *simple*

in Section 3.2: the number of relations is 11 and there is one cluster structure for each relation, representing an expansion by the means of a single relation.

$$\binom{n}{k} = \binom{11}{1} = 11 \; combinations$$

For the case $k = 2$ the number of combinations is:

$$\binom{n}{k} = \binom{11}{2} = 55 \; combinations$$

and the possible combinations of the semantic relations (according to the abbreviations shown in Table 1 and Table 2) are:

SYN TRG
SPC SYN, SPC TRG
PAR SPC, PAR SYN, PAR TRG
JJB PAR, JJB SPC, JJB SYN, JJB TRG
JJA JJB, JJA PAR, JJA SPC, JJA SYN, JJA TRG
GEN JJA, GEN JJB, GEN PAR, GEN SPC, GEN SYN, GEN TRG
COM GEN, COM JJA, COM JJB, COM PAR, COM SPC, COM SYN, COM TRG
BGB COM, BGB GEN, BGB JJA, BGB JJB, BGB PAR, BGB SPC, BGB SYN, BGB TRG
BGA BGB, BGA COM, BGA GEN, BGA JJA, BGA JJB, BGA PAR, BGA SPC, BGA SYN, BGA TRG
ANT BGA, ANT BGB, ANT COM, ANT GEN, ANT JJA, ANT JJB, ANT PAR, ANT SPC, ANT SYN, ANT TRG

The cases $k = 3, ..., k = 11$ follow the same structure and generate $\binom{11}{3} = 165$, $\binom{11}{4} = 330$, $\binom{11}{5} = 462$, $\binom{11}{6} = 462$, $\binom{11}{7} = 330$, $\binom{11}{8} = 165$, $\binom{11}{9} = 55$, $\binom{11}{10} = 11$ and $\binom{11}{11} = 1$ combinations. The latter set is the more complex, describing the combination:

ANT, BGA, BGB, COM, GEN, JJA, JJB, PAR, SPC, SYN, TRG

that represents the combination of all the considered semantic expansions.
To achieve such result a fully automatic procedure providing all the possible combinations of the semantic relations obtained in Section 3.2 has been produced and made run in parallel to keep the execution performances attainable.
The lists produced in Section 3.2, representing the categories labels expanded through the 11 semantic relations available on *WordNet*, *Onelook dictionaries* and *Google Books N-grams* are loaded and kept separated by semantic relation type. Each list of the target corpus is built combining the documents of the corpus described in Section 3.2 in k ways: for each k, all the 1565 lists corresponding to the categories labels are expanded combining their contents by different semantic relations, giving as result a folder named as the considered relations (e.g. `ANT`, `ANT SYN`, etc.). Each folder contains the same hierarchy and number of files as the original folders, while the documents within the folder contain the combination of the original lists. To avoid category names replication within the documents, the labels of the categories have been excluded from the computation and added at the end of the process. After the combination process the files are processed by the `enhanceLists` and `getLeavesToSort` libraries described in Section 3.3 to gain coherence with the resources generated in Section 3.2 and the assessment which is proposed in Section 4.1.
As a result of the procedure 11 folders have been generated, each corresponding to a configuration of k from 1 to 11. Each folder contains the set of all the possible combinations of 11 semantic relations in k ways as directories, each of which preserves the original cluster hierarchy maintaining the macro-category labels as folder names and the category labels as file names. The procedure is applied to the use-case taxonomy and generates a total of 2047 combinations of 11 semantic relations, one for each extraction of k elements, for a total of 3.203.555 generated files. The generation and the structure of the above described resources is depicted in Figure 7.

4. Evaluation

The assessment is realized making use of a GATE pipeline which is used to compute a set of user queries, processing them with different corpora loaded as gazetteers. The pipeline is integrated into a pseudo-relevance search engine which allows to automatically get from the user query the information needed to rank the Yelp taxonomy labels, reformulating the actual query as the most ranked labels respect to the query terms. The aim of the assessment is to evaluate the impact of each semantic relation in term expansion and to highlight the strengths and weaknesses in employing different combinations of such relations. The evaluation tools are described in Section 4.1 and the criteria of the assessment are discussed in Section 4.2. An evaluation of the impact of each expansion is proposed in Section 4.3 and is compared with other evaluations made through the combination of resources.

4.1. Evaluation infrastructure

The evaluation has been assessed processing the query terms by the GATE engine. The default ANNIE pipeline has been employed, which is composed by the ANNIE tokenizer, the RegEx sentence splitter, the TreeTagger POS-tagger and the ANNIE NE JAPE transducer; the corpora which are described in Section 3.2 and in Section 3.3 have been loaded as gazetteers (Figure Figure 8).

This approach combines term filtering based on part-of-speech tags and cluster labels expansion based on offline building of gazetteers. Such approach is flexible enough to

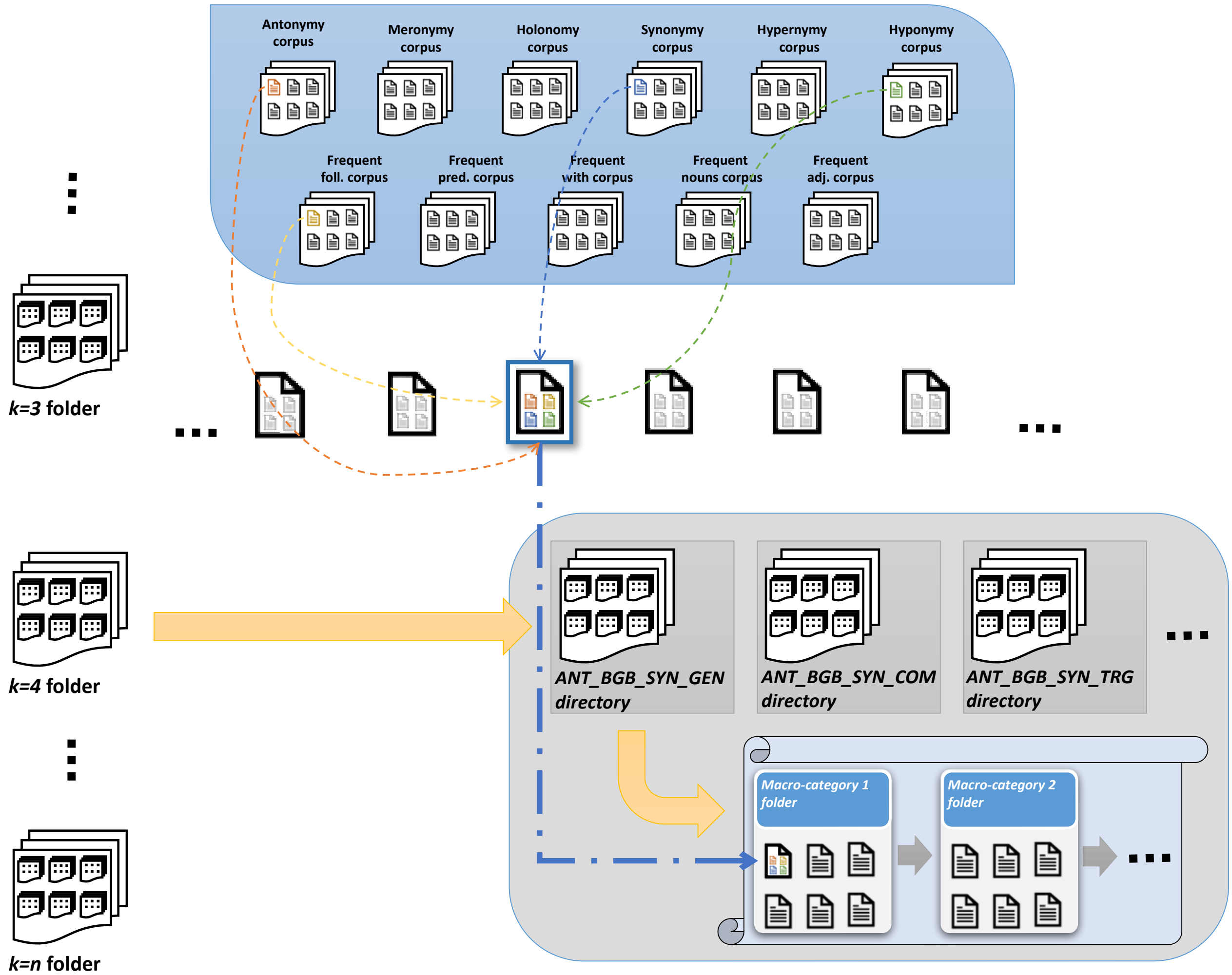

Figure 7: Generation and structure of the combined corpora

allow the comparison of different strategies for building expansion corpora.

The assessment is focused on query reformulation and consists in three phases:

1. **Query cleaning.** The query terms are pruned from stop words, conjunctions, articles etc. and lemmatized through the GATE framework;
2. **Browsing the corpus.** The extracted query terms are used as search terms browsing the loaded configuration of the corpus. The occurrence of a query term within a list increases the relevance score of the list respect to the user query. If a term is present into a list by multiple semantic relations, this reflects in more significance given to the term respect to the taxonomy label. The sum of the weights constitutes the score of the list and is compared to the rank of the other lists;
3. **Query reformulation.** The query is reformulated as the label(s) with the majority voting of the scores obtained in the second phase. The elected pseudo-queries are the top equally ranked category labels (at most 5), respect to the sum of weights of the query terms processed in the first phase. It is to be noticed that using the T-box of an ontology as target taxonomy, the elected labels can be immediately used to query the underlying A-box and retrieve the actual data from the knowledge base.

Each corpus described in Section 3.3 is separately loaded in the pipeline as gazetteers to obtain the evaluation. The taxonomy categories which are included in the corpus described in the first phase of Section 3.2 are loaded into the pipeline to provide a bottom line to compare with the expansions. After setting the evaluation for the bottom line, the folders which have been generated through the `generateListsBySemRel` function (corresponding to a single relation expansion) are included in the pipeline one at a time and evaluated separately. Afterwards, to compare the combination of each semantic relation with the bottom line and the single relation expansions, the assessment is replicated for each combination folder described in Section 3.3 (corresponding to a combination of the 11 relations).

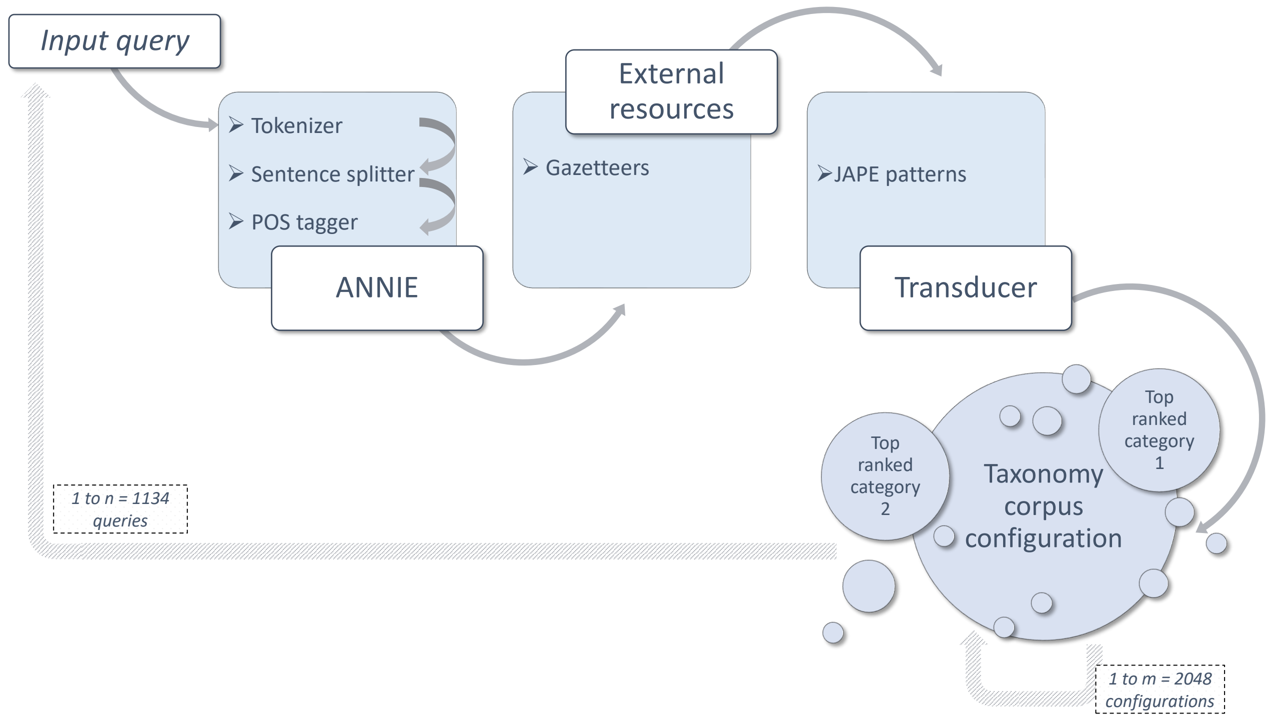

Figure 8: Default ANNIE pipeline loaded with different configurations of corpora

4.2. Evaluation criteria

The evaluation of the estimated intents is assessed using the classic information retrieval metrics of precision, recall, F-measure and accuracy. The precision metric represents the capabilities of the pseudo-relevance search engine to retrieve relevant documents, which are the taxonomy categories, respect to the user query. The recall metric is exploited to measure the capabilities of the pipeline in retrieving all the relevant documents which can be retrieved by the system. The F-measure metric is the expression of the harmonic mean of precision and recall, while accuracy quantifies the capacity of the classifier to match expected results. The above described metrics are calculated counting the number of True Positives (TP), False Positives (FP), False Negatives (FN) and True Negative (TN), which are considered as follows:

- A *true positive* is considered when a taxonomy category which is present in the ground truth is also present in the GATE pipeline outcome;
- a *false positive* is considered when none of the elected categories is present within the ground truth;
- a *true negative* is considered when the elected categories set is empty and the ground truth is empty as well;
- a *false negative* is considered when the elected categories set is empty, while the ground truth is not.

$$\textbf{Precision} = \frac{TP}{TP + FP} \qquad \textbf{Recall} = \frac{TP}{TP + FN}$$

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\textbf{F-measure} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

The graph of the Receiver Operating Characteristic (**ROC**) curve has been also evaluated. The ROC curve is plotted on the specificity and sensitivity axes, representing the false positives and true positives rates. Sensitivity and specificity are respectively calculated summing the number of TPs at k and the number of FPs at k, where k varies between 1 and the number of queries, divided by the total of TPs and FPs, respectively. The plot of the curve gives a graphic overview of the classification capabilities of the assessment pipeline using the loaded set of resources, which is different for each evaluation. The Area Under the Curve (**AUC**) is evaluated for each classifier to express its performances calculating the area under the ROC curve. The AUC analysis is performed to provide a numeric comparison of the capabilities of the classifier loaded with a different configuration of resources. To derive the above described measures as an expression of the impact of each semantic relation in term expansion, the code of the assessment infrastructure has been set to loop among a set of queries and calculating the metrics by counting TPs, FPs, TNs and FNs.

The queries which have been evaluated are a portion of the AOL Query Logs [Pass et al. (2006)], which contain more than 36 million questions made by users on the popular search engine. The set has been reduced to 1134 queries, filtering the questions in the form: *"Where can I find. . . ?"*, *"Where can I buy. . . ?"*, *"Where can I get. . . ?"*, *"Where can I purchase. . . ?"*, which is the sub-set of the transactional queries [Jansen et al. (2008)] of the AOL Query Logs. This operation has been made to make the test queries domain the

Table 4
Performances of a simple classifier loaded with one relation expansion

Relation name	Precision values	Recall values	F-measure values	Accuracy values
BASELINE	*62,52%*	*29,82%*	*40,18%*	*25,95%*
ANT	60,47%	29,77%	39,90%	25,63%
BGA	48,15%	40,23%	43,84%	28,64%
BGB	42,86%	45,21%	44,00%	28,72%
COM	59,92%	29,79%	39,80%	25,55%
GEN	55,30%	31,96%	40,51%	26,11%
JJA	49,78%	38,12%	43,18%	28,16%
JJB	46,94%	36,26%	40,91%	26,42%
PAR	60,50%	29,99%	40,10%	25,79%
SPC	51,44%	33,93%	40,89%	26,34%
SYN	**62,82%**	41,07%	49,67%	33,62%
TRG	47,38%	**64,12%**	**54,49%**	**37,90%**

same as the use-case domain, which is the field of POIs. The ground truth has been obtained as an annotation of the evaluation queries with the corresponding Yelp taxonomy labels. This operation has been performed by 5 different assessors and the ground truth for each query has been elected as the most frequent annotation among them. The evaluations are shown in the next section.

4.3. Assessment and discussion

The metrics are initially calculated loading the baseline resources (Section 3.2.1) to evaluate the retrieval capabilities of the system as a pattern matching engine. The same operation has been done loading the single-relation expansions corpora (Section 3.2) as resources, quantifying the most impacting relation among those which can be employed to provide additional knowledge for retrieval. A third step has been made, loading as resources different combinations of the produced corpora (Section 3.3), highlighting the most effective combination of the semantic relations. All the evaluations are made through the techniques discussed in Section 4.2.

Baseline evaluation. To provide a bottom line for the retrieval system capabilities the assessment engine has been configured as a pattern matching engine filling the cluster lists with the taxonomy labels only, each with unitary weight; the queries have been tokenized and lemmatized. The results are shown in Figure 9 ~ Figure 19, depicted as a straight line expressing the bottom line for each evaluation which is discussed below.

Single semantic expansion evaluation. The evaluation has been provided to highlight the different impact of different relations when building a semantic search system based on semantic expansion. The assessment has been made loading the gazetteers which have been produced expanding the cluster labels, one at a time. The results are shown in Table 4 and in Figure 9. This evaluation quantifies the improvement of each cluster label expansion made by the means of different available semantic relations respect to a pattern matching system.

This evaluation shows that is held more precision loading the pattern matching classifier with the synonymy expansion (SYN) of the taxonomy label and the highest recall is reached loading the TRG expansion (words that are statistically associated with the query word in the same piece of text). The highest values for F-measure and accuracy are both associated to the TRG expansion, probably because the set of queries is made by real users following similar patterns when making questions on specific fields. Respect to the bottom line the precision measure does not represent a significant improvement, probably because the simple pattern matching performs well in the case of the AOL queries, which include most of the words which are also in the labels of the well detailed Yelp taxonomy. The F-measure and accuracy measures hit a significant improvement respect to the baseline, showing the best result for the classifier loaded with the TRG expansion corpus; in particular, an improvement of the F-measure reflects in an improvement in retrieving documents with an NLP system. It is to be noticed that the baseline classifier is the simplest possible, exploiting tokenization and pattern matching only, and each expansion corpus contains the lemmatized taxonomy labels without expansion; including a selection of related words to the baseline (e.g. through Babelnet [Navigli and Ponzetto (2012)])

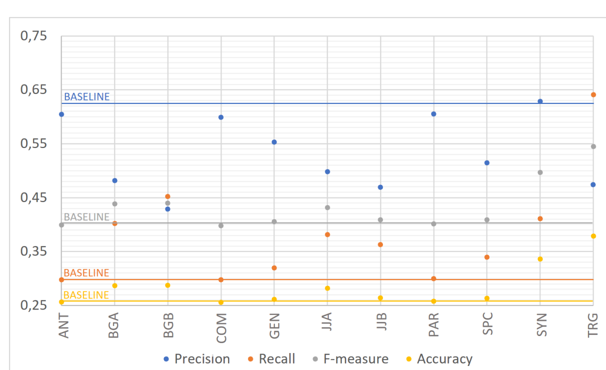

Figure 9: Single semantic relation evaluation plot (*k=1* set)

or, as an example, the lemmas of the words which may be contained in the documents to retrieve, would provide significantly higher results.

Combination of the semantic relations. A further experiment has been assessed evaluating all the combinations of the produced corpora as described in Section 3.3: the terms contained in the gazetteers have been combined to produce a set of documents for each combination of k elements, with k ranging from 1 to 11, which is the total number of semantic relations. To provide the analyses which are depicted in Figure 9 ~ Figure 19, the data obtained from the evaluations described in Section 3.3 has been employed. The data is summarized in a table composed by the number of configuration and the considered semantic relations; for each configuration the total TP, FP, TN and FN has also been included, together with precision, recall, F-measure and accuracy. The dataset has been ordered sorting the configurations by relation name respect to each cardinality subset ($k = 1$, $k = 2$,...); each configuration label is associated to an id which is used as x-axis value to simplify the representation of the graphs (e.g. in Figure 10 the configuration `ANT BGA` corresponds to the x-value 13 respect to the subset k=2, since it is the 13th relation in such set). This step has been performed to better highlight the best performing classifier among the others respect to the same cardinality class. The precision, recall, F-measure and accuracy values have been used as y-axis to draw the scatter plots. The overall comparison of the best performing configurations is presented in Table 4, showing the highest values found for each configuration respect to the complete set. The full assessment is available for research purposes at: https://github.com/lmassai/semRels/eval, containing the labels of each configuration, the relative id number and the reference set, together with the metrics evaluations. To better highlight the improvements of each configuration respect to the baseline pattern matching, the baseline result obtained for each measure has been reported on the graphs as a straight horizontal line corresponding to the precision, recall, F-measure and accuracy of the baseline classifier.

To report the most representative information extracted from the complete dataset analysis, corresponding to the best performances of each classifier, the higher values obtained through the evaluation are reported in Table 5, as an excerpt of the full assessment. This data has been used to draw the graphs in Figure 9 ~ Figure 19, showing graphically the measurements for the 10 better performing classifiers, respect to each measure considered. The overall best (in bold) precision value is held by the classifier loaded with the combination of `ANT`, `SYN`, and more in general by all those configurations including them and the PAR expansion, which are the ones which better perform as single relations in the previous analysis (Table 4). The higher results held by the other combinations in Table 5 are probably due to

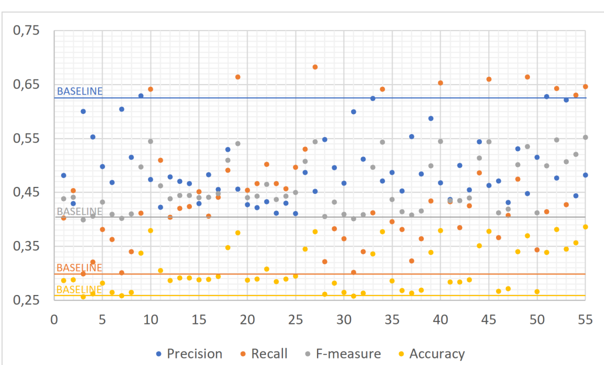

Figure 10: Configurations in (k=*2* set)

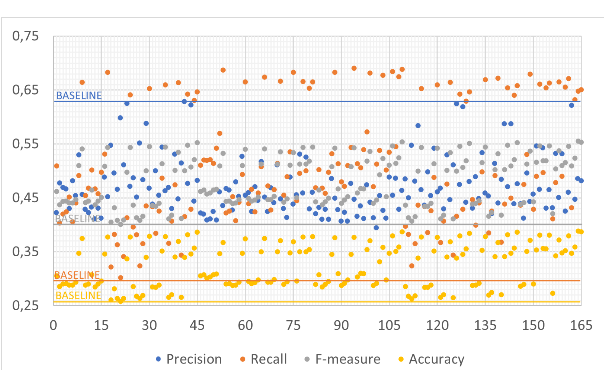

Figure 11: Configurations in (k=*3* set)

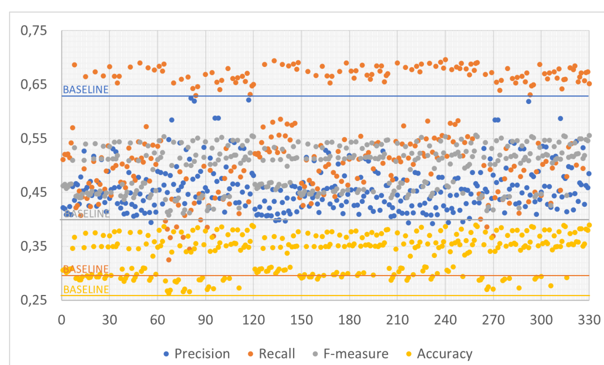

Figure 12: Configurations in (k=*4* set)

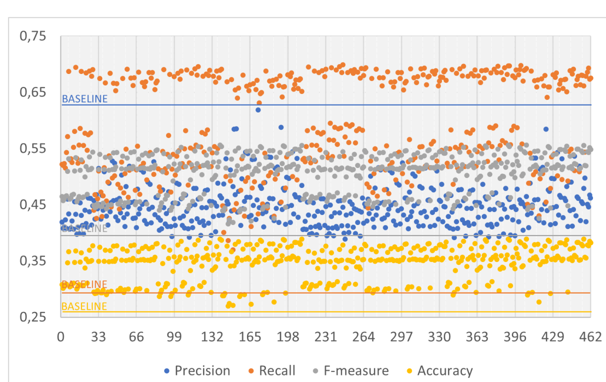

Figure 13: Configurations in (k=*5* set)

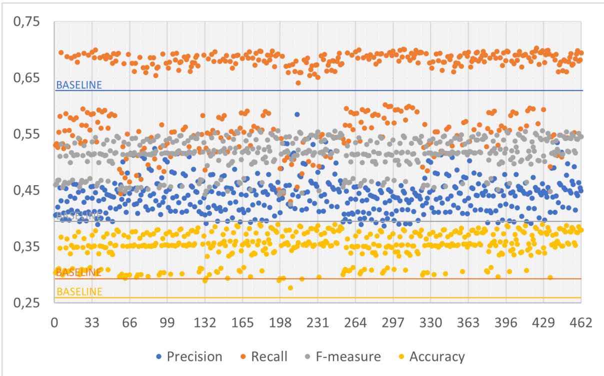

Figure 14: Configurations in (*k=6* set)

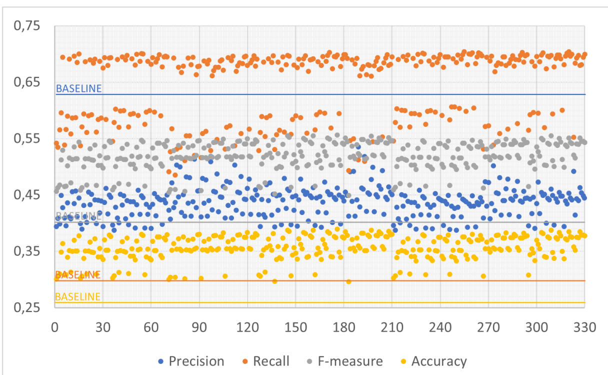

Figure 15: Configurations in (*k=7* set)

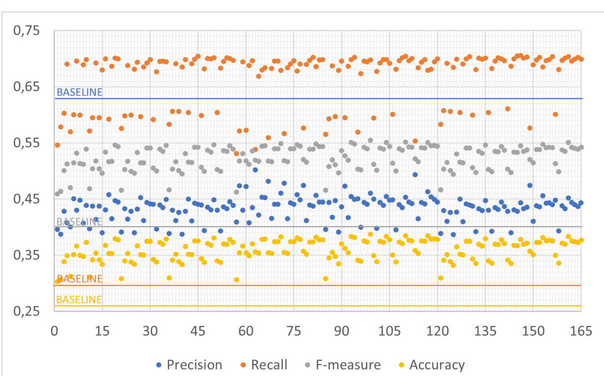

Figure 16: Configurations in (*k=8* set)

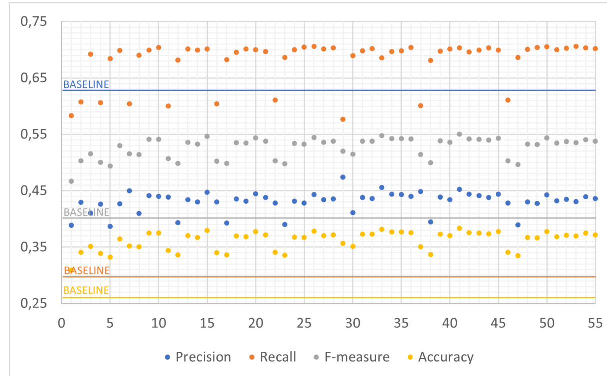

Figure 17: Configurations in (*k=9* set)

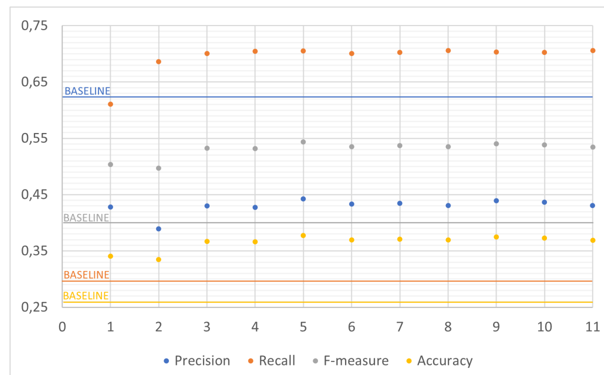

Figure 18: Configurations in (*k=10* set)

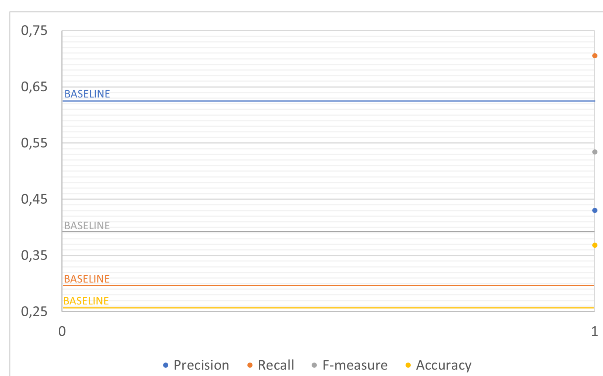

Figure 19: Configurations in (*k=11* set)

the presence of such relations within the combination and the slightly lower performances of those classifiers is due to some query drift introduced by other relations lowering the performances. The best recall result is achieved employing the `BGA BGB GEN JJA JJB PAR SPC SYN TRG` configuration, showing higher values with more relations included; almost all of the 10 top results overcome the results obtained with any single relation expansion. The F-measure evaluations show that the best results are obtained by the `BGB GEN PAR SYN TRG` configuration which, despite the evident improvement respect to the baseline, is not far by the other measurements and this is also the case for the accuracy evaluations. The accuracy evaluations are headed by the `PAR SPC SYN TRG` configuration; all the registered accuracy values show very low performances achieved by the classifiers and the main reason is that the accuracy of a classifier is tightly bound to the classifier core retrieval capabilities which are, in this case, naïve; this is noticeable looking at the accuracy value of the bottom line evaluation. The reason for this is that the main target of the study is to highlight the *improvements* in employing different semantic relations in any classifier through semantic expansion of the cluster labels: the results are noticeable

Table 5
Highest values for cardinality set

Set	Max precision values		Max recall values		Max F-measure values		Max accuracy values	
k=1	SYN	62,82%	TRG	64,12%	TRG	54,49%	TRG	37,89%
k=2	**ANT, SYN**	**62,87%**	BGB, TRG	68,26%	SYN, TRG	55,24%	SYN, TRG	38,60%
k=3	ANT, PAR, SYN	62,85%	BGB, GEN, TRG	69,02%	PAR, SYN, TRG	55,49%	PAR, SYN, TRG	38,84%
k=4	ANT, COM, PAR, SYN	62,48%	BGB, GEN, SYN, TRG	69,60%	BGB, PAR, SYN, TRG	55,63%	**PAR, SPC, SYN, TRG**	**38,92%**
k=5	ANT, COM, PAR, SPC, SYN	61,90%	BGA, BGB, GEN, SYN, TRG	69,94%	**BGB, GEN, PAR, SYN, TRG**	**55,63%**	GEN, PAR, SPC, SYN, TRG	38,92%
k=6	ANT, COM, GEN, PAR, SPC, SYN	58,53%	BGB, GEN, JJA, SPC, SYN, TRG	70,26%	BGB, GEN, PAR, SPC, SYN, TRG	55,63%	BGB, GEN, PAR, SPC, SYN, TRG	38,92%
k=7	ANT, COM, GEN, JJA, PAR, SPC, SYN	53,49%	BGA, BGB, GEN, JJA, SPC, SYN, TRG	70,43%	ANT, BGB, GEN, PAR, SPC, SYN, TRG	55,63%	ANT, BGB, GEN, PAR, SPC, SYN, TRG	38,92%
k=8	ANT, BGA, COM, GEN, JJA, PAR, SPC, SYN	50,17%	BGA, BGB, GEN, JJA, PAR, SPC, SYN, TRG	70,57%	ANT, BGB, COM, GEN, PAR, SPC, SYN, TRG	55,46%	ANT, BGB, COM, GEN, PAR, SPC, SYN, TRG	38,76%
k=9	ANT, BGA, COM, GEN, JJA, JJB, PAR, SPC, SYN	47,41%	**BGA, BGB, GEN, JJA, JJB, PAR, SPC, SYN, TRG**	**70,61%**	ANT, BGB, COM, GEN, JJA, PAR, SPC, SYN, TRG	55,04%	ANT, BGB, COM, GEN, JJA, PAR, SPC, SYN, TRG	38,37%
k=10	ANT, BGA, BGB, COM, GEN, JJA, PAR, SPC, SYN, TRG	44,24%	ANT, BGA, BGB, GEN, JJA, JJB, PAR, SPC, SYN, TRG	70,61%	ANT, BGA, BGB, COM, GEN, JJA, PAR, SPC, SYN, TRG	54,37%	ANT, BGA, BGB, COM, GEN, JJA, PAR, SPC, SYN, TRG	37,73%
k=11	ANT, BGA, BGB, COM, GEN, JJA, JJB, PAR, SPC, SYN, TRG	43,00%	ANT, BGA, BGB, COM, GEN, JJA, JJB, PAR, SPC, SYN, TRG	70,57%	ANT, BGA, BGB, COM, GEN, JJA, JJB, PAR, SPC, SYN, TRG	53,44%	ANT, BGA, BGB, COM, GEN, JJA, JJB, PAR, SPC, SYN, TRG	36,86%

looking at the difference of each evaluation respect to the bottom lines (in Table 4), giving an indication on which combinations enhance the most the basic retrieval capabilities of the classifier and how much.

ROC and AUC analysis. The metrics which have been calculated so far are useful to perform the analysis of the receiver operating characteristic (ROC) curve and consequently of the area which lays over it, the Area Under the curve, or AUC. The ROC analysis allows to get an overview of the retrieval capabilities of a classifier, showing also the configuration which represents the best tradeoff between accuracy and query drift through the AUC analysis. The AUC has been calculated for each of the 2047 classifiers and the results are presented for the single relation expansions and for the 10 higher AUC classifiers. A discussion has been also presented for the 10 minimum AUC classifiers.

The single semantic relation which is resulted to be the most effective to improve the retrieval capabilities of a classifier, showing the overall best performances among a large set of queries, is the antonymy (ANT) relation, having the highest value of AUC with the given classifier. By the nature of the measure, this relation represents also the best tradeoff between precision and recall evaluations seen in Table 4 for single relations. The values are obtained through the analysis of the ROC curve for each single relation and are shown in Figure 20.

The 10 highest and lowest AUCs among the evaluations of all the classifiers can be found in Table 7 and Table 8, showing that the configuration with the wider area is ANT GEN SPC, while the lowest is obtained using BGA COM JJA JJB PAR SPC TRG. The configurations are ordered by the highest and lowest AUC value of each classifier, loaded with the configuration of resources in the first column.

Table 6
AUC values for single relation configurations

Relation name	AUC
ANT	**0,524839**
PAR	0,523924
GEN	0,523407
SPC	0,522862
COM	0,515860
BGA	0,515306
SYN	0,513280
BGB	0,512592
JJA	0,510378
JJB	0,508958
TRG	0,508461

It is to be noticed that the PAR relation, which holds a better AUC value than both GEN and SPC in the single relation evaluation which is shown in Table 6 is not included in the best configuration of the present evaluation. The reason for this is probably that it introduces some query drift when used together with ANT and GEN lowering the performances of the configuration, while SPC does not. Considering the cardinality of the configurations which are included in Table 7 it can be seen that higher values are mainly reached when less semantic relations are involved, while the lowest values, which

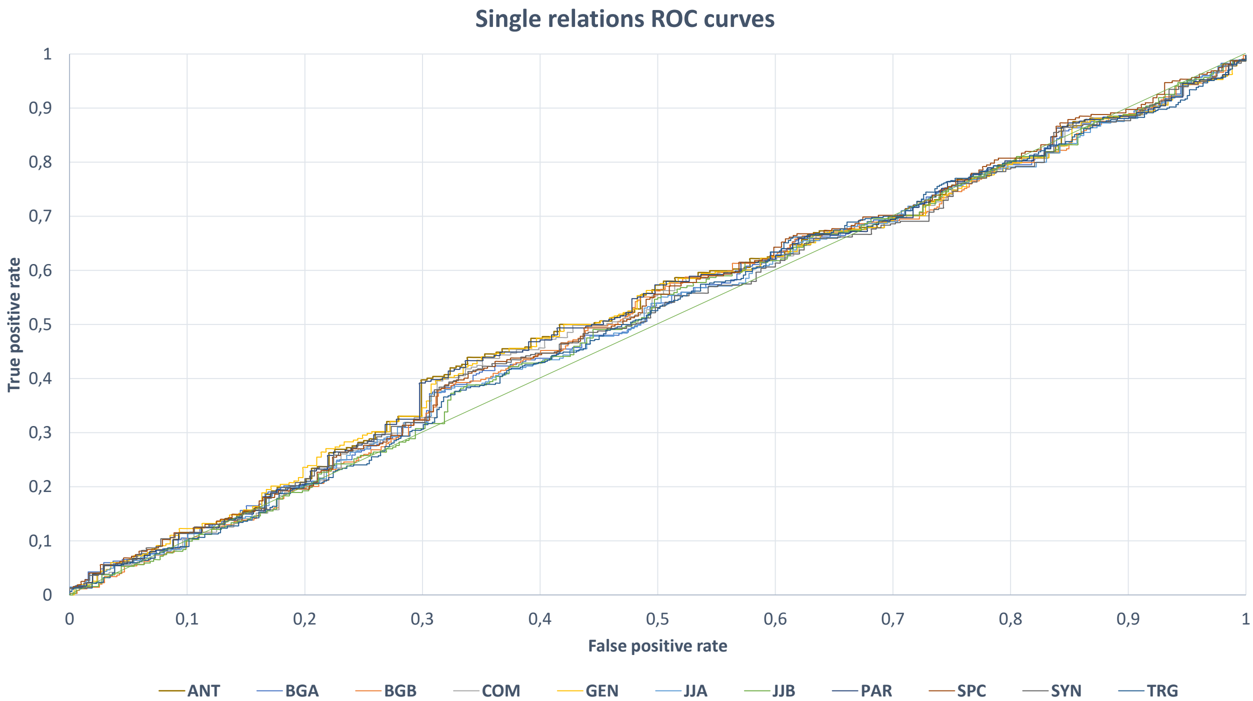

Figure 20: Single relations ROC analysis

Table 7
Overall highest AUC values

Relation name	Max AUC
ANT GEN SPC	**0,525342**
ANT PAR	0,524842
ANT	0,524839
GEN SPC	0,524411
ANT GEN	0,524361
ANT GEN PAR SPC	0,524181
PAR	0,523924
ANT SPC	0,523922
ANT GEN PAR	0,523531
GEN	0,523407

Table 8
Overall lowest AUC values

Relation name	Min AUC
BGA COM JJA JJB PAR SPC TRG	**0,475563**
ANT BGA COM JJA JJB PAR SPC TRG	0,475563
BGA JJA JJB PAR SPC TRG	0,476426
ANT BGA JJA JJB PAR SPC TRG	0,476426
BGA COM JJA JJB PAR SYN TRG	0,477172
ANT BGA COM JJA JJB PAR SYN TRG	0,477172
ANT COM JJA JJB PAR SYN TRG	0,477392
BGA COM GEN JJA JJB PAR SPC TRG	0,478083
ANT BGA COM GEN JJA JJB PAR SPC TRG	0,478083
ANT BGB COM JJA JJB PAR SYN TRG	0,478106

are included in Table 8, show some of the longest combinations. Such configurations have to be avoided for semantic expansions since their employment leads to a worsening of the performances of the classifier, as it is noticeable by comparison with the baseline. The full assessment is available for research purposes at:
https://github.com/lmassai/semRels/evalCombinations.

5. Conclusions

In this paper many concepts in the field of automatic term expansion have been discussed; the fields of relevance feedback and search results clustering have been investigated in detail. Automatic term expansion often deals with the latent semantics of the terms and knowledge resources like WordNet, the Onelook dictionaries and the Google Books N-grams databases are used to get the latent senses. The relations which are accessible through such repositories have been exploited using a wide taxonomy as input of an automatic procedure aimed at semantically expanding the cluster labels. The taxonomy labels have been converted in lists and their labels have been expanded by the means of each semantic relation, then a further step has been made to combine the extracted resources. To obtain a comparison of the effects in employing the different assets in document retrieval, the generated resources have been integrated as gazetteers into several GATE pipelines. The pipelines have been used as separate pseudo-relevance systems to generate pseudo-queries corresponding to the taxonomy labels. The evaluation gives evidence of the most effective combination of the semantic relations in term expansion and shows the best trade-off between accuracy and noise introduction, highlighting the

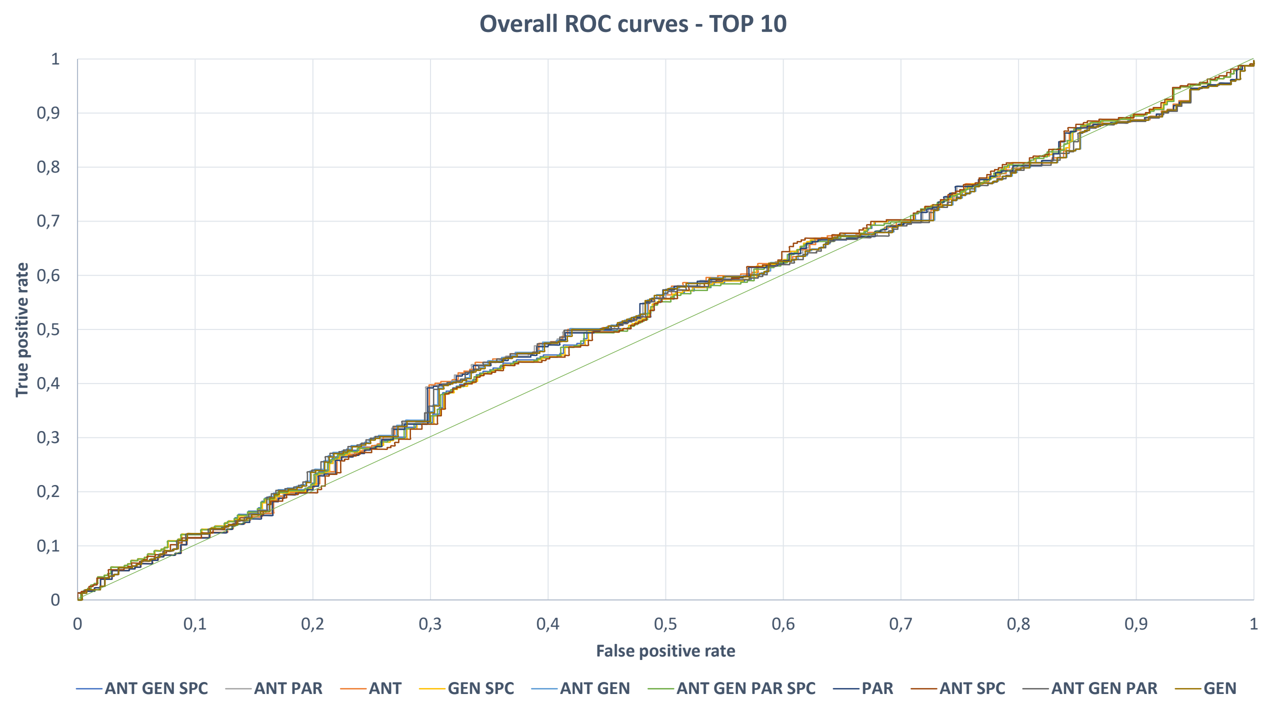

Figure 21: Overall ROC analysis(highest 10)

different impact in employing each relation and each combination of them for estimating the intent of a user querying any search engine with natural language. Based on such results, this study demonstrates which combinations are better suited for improving the most recent search engine's capabilities. The generation and the combination procedures are completely automatic and provide high scalability: the system can be input with any taxonomy and it can be applied to any structured domain like indexes, menu structures, ontology taxonomies, etc. The generated resources have demonstrated to be easily exploitable to reformulate any natural language query as the labels of an input taxonomy, enhancing the retrieval of any data underneath and using the most effective combination of the available semantic relations.

6. Acknowledgements

A special thanks is due to Yelp (https://www.yelp.com/) and AOL (https://www.aol.com/) organizations who made available the huge amount of data which is exploited to build the structure of the corpora for the use-case evaluation and the queries which have been used in the assessment. Most of the data which is exploited to compute such resources is obtained through the WordNet [Fellbaum (1998)] framework, made available by the Princeton University which is kindly acknowledged. Without these data and resources, the ideas described in this work couldn't have been provable.

References

Azad, H.K., Deepak, A., 2019. Query expansion techniques for information retrieval: A survey. Inf. Process. Manag. 56, 1698–1735. URL: http://dblp.uni-trier.de/db/journals/ipm/ipm56.html#AzadD19.

Baeza-Yates, R.A., Ribeiro-Neto, B.A., 1999. Modern Information Retrieval. ACM Press / Addison-Wesley. URL: http://www.ischool.berkeley.edu/~hearst/irbook/glossary.html.

Bai, J., Song, D., Bruza, P., Nie, J.Y., Cao, G., 2005. Query expansion using term relationships in language models for information retrieval., in: Herzog, O., Schek, H.J., Fuhr, N., Chowdhury, A., Teiken, W. (Eds.), CIKM, ACM. pp. 688–695. URL: http://dblp.uni-trier.de/db/conf/cikm/cikm2005.html#BaiSBNC05.

Bhogal, J., MacFarlane, A., Smith, P., 2007. A review of ontology based query expansion. Inf. Process. Manag. 43, 866–886. URL: http://dblp.uni-trier.de/db/journals/ipm/ipm43.html#BhogalMS07.

Clink, K., 2007. Onelook: Dictionary search. Reference Reviews 21, 35–35. doi:https://doi.org/10.1108/09504120710838912.

Cortes, C., Vapnik, V., 1995. Support vector networks. Machine Learning 20, 273–297.

Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., 2002. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications, in: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02).

Datamuse, 2016. Resource available online. URL: http://www.datamuse.com/api/.

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

Efthimiadis, E.N., 2000. Interactive query expansion: A user-based evaluation in a relevance feedback environment. J. Am. Soc. Inf. Sci. 51, 989–1003. URL: https://doi.org/10.1002/1097-4571(2000)9999:9999%3C::AID-ASI1002%3E3.0.CO;2-B, doi:10.1002/1097-4571(2000)9999:9999\%3C::AID-ASI1002\%3E3.0.CO;2-B.

Fellbaum, C. (Ed.), 1998. WordNet: an electronic lexical database. MIT Press.

Jansen, B.J., Booth, D.L., Spink, A., 2008. Determining the informational, navigational, and transactional intent of web queries. Inf. Process. Manag. 44, 1251–1266. URL: https://doi.org/10.1016/j.ipm.2007.07.015, doi:10.1016/j.ipm.2007.07.015.

Kurland, O., Lee, L., Domshlak, C., 2006. Better than the real thing?

iterative pseudo-query processing using cluster-based language models. CoRR abs/cs/0601046. URL: http://arxiv.org/abs/cs/0601046, arXiv:cs/0601046.

Lloyd, S.P., 1982. Least squares quantization in PCM. IEEE Trans. Inf. Theory 28, 129–136. URL: https://doi.org/10.1109/TIT.1982.1056489, doi:10.1109/TIT.1982.1056489.

Lopes, L., Gadge, J., 2014. Hybrid approach for query expansion using query log. International Journal of Applied Information Systems 7, 30–35. Published by Foundation of Computer Science, New York, USA.

Mitra, M., Singhal, A., Buckley, C., 1998. Improving automatic query expansion, in: Croft, W.B., Moffat, A., van Rijsbergen, C.J., Wilkinson, R., Zobel, J. (Eds.), SIGIR '98: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 24-28 1998, Melbourne, Australia, ACM. pp. 206–214. URL: https://doi.org/10.1145/290941.290995, doi:10.1145/290941.290995.

Navigli, R., Ponzetto, S.P., 2012. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artif. Intell. 193, 217–250. URL: https://doi.org/10.1016/j.artint.2012.07.001, doi:10.1016/j.artint.2012.07.001.

Pass, G., Chowdhury, A., Torgeson, C., 2006. A picture of search, in: Jia, X. (Ed.), Proceedings of the 1st International Conference on Scalable Information Systems, Infoscale 2006, Hong Kong, May 30-June 1006, ACM. p. 1. URL: https://doi.org/10.1145/1146847.1146848, doi:10.1145/1146847.1146848.

Rocchio, J.J., 1971. Relevance feedback in information retrieval, in: Salton, G. (Ed.), The Smart retrieval system - experiments in automatic document processing. Englewood Cliffs, NJ: Prentice-Hall, pp. 313–323.

Schmid, H., 1994. Probabilistic part-of-speech tagging using decision trees, in: Proceedings of the International Conference on New Methods in Language Processing, Manchester, UK.