

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/271726222>

# Ensembling Classifiers for Detecting User Intentions behind Web Queries

Article in IEEE Internet Computing · January 2015

DOI: 10.1109/MIC.2015.22

---

CITATIONS

3

---

READS

208

1 author:



[John Atkinson-Abutridy](#)

Universidad Adolfo Ibáñez

78 PUBLICATIONS 756 CITATIONS

[SEE PROFILE](#)

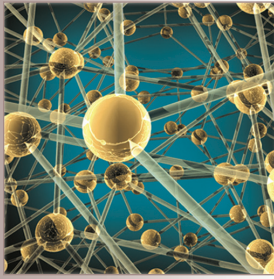
Some of the authors of this publication are also working on these related projects:



An effective Linguistically-motivated computational model for opinion retrieval in sentiment analysis tasks [View project](#)



Innovando la Innovación [View project](#)



# Ensembling Classifiers for Detecting User Intentions behind Web Queries

Discovering user intentions behind Web search queries is key to improving user experience. Usually, this task is seen as a classification problem, in which a sample of annotated user query intentions are provided to a supervised machine learning algorithm or classifier that learns from these examples and then can classify unseen user queries. This article proposes a new approach based on an ensemble of classifiers. The method combines syntactic and semantic features so as to effectively detect user intentions. Different setting experiments show the promise of this linguistically motivated ensembling approach, by reducing the ranking variance of single classifiers across user intentions.

**Alejandro Figueroa**  
Yahoo Research, Universidad  
Diego Portales, and Universidad  
Andres Bello, Santiago, Chile

**John Atkinson**  
Universidad de Concepción,  
Concepción, Chile

In recent years, the Web has not only become a huge repository of information, but also a place where people can interact and access different kinds of resources such as services and applications. However, there's a gap between user needs and the resources available to meet them. Users express their requests by entering a short sequence of query terms, which are further interpreted by search engines to provide relevant answers. This makes search engines key players in understanding and efficiently resolving hundreds of millions of queries per day.

To get valid query interpretations, a main step involves discriminating the user's intention, which varies from fulfilling information needs to using search engines as navigational tools to reach

specific websites. Search engines can also perform transactions by providing access to different resource types including maps, lyrics, and books. Automatically detecting user's intentions is a key challenge for search engines as they can improve user's experience by getting more useful results and tailoring them to their specific needs. On the one hand, the intention of some highly frequent queries (for example "wikipedia" and "yahoo") can easily be identified by benefiting from any type of hash table extracted from analyzing click patterns across search logs. Furthermore, a user's intention behind queries with a limited set of patterns (that is, "term1 term2 lyrics" and "define term1 term2"), can also be readily recognized. Nevertheless, it's difficult to determine the intention of

## Related Work in Detecting User Intent behind Web Search Queries

Some studies have proposed a taxonomy for Web search engine queries based on manual inspections.<sup>1</sup> A first level consists of three canonical branches, which cover most user goals when searching: navigational (for example, “facebook” and “twitter”), information-oriented (for example, “how do I get rid of acne?” and “obama bio”), and resource-oriented (for example, “berlin map” and “free anti-virus”).

Current approaches to automatically labeling and classifying search queries randomly select instances extracted from search logs.<sup>2</sup> They aim to discover relevant features to discriminate one intention from the other, which includes keywords and information extracted from the pages visited by users. Resulting navigational queries were found to be classified by organization’s and people’s names (for example, “dell” and “madonna”) and domain suffixes (for example, “.com”). On the other hand, resource queries are short and likely to contain keywords such as lyrics, movies, recipes, and images (for example, “lentil soup recipe” and “justin bieber images”), whereas informational queries are longer, and usually formulated with question words resembling natural language text (for example, “what is the biggest organ in the human body?”).

Other methods group Web queries based on these three canonical segments using *k*-means clustering and a feature-rich representation. Each item in the search log comprises features such as user identification, cookie, time of day, query terms, and type of content collection for which the user is searching. In addition, each item was enriched with the query length, a number modeling the search engine results page visited during a given interaction, and the number of times a user changed the query during a session. The method then assigns terms to each record, such as informational, navigational, or transactional.<sup>2</sup>

Statistical language models have also been exploited to classify Web query instances based on their intention.<sup>2</sup> These instances are then used to automatically categorize new queries via exact terms’ matching. However, the approach is too restrictive because it matches frequent elements. To deal with this issue, intention-classification approaches use support vector machines (SVMs) and Naive Bayes classifiers,<sup>3</sup> showing that an SVM obtained better results on the informational category,

whereas Naive Bayes did well for the other two types of intentions. Experiments indicate that word-based features become key in recognizing resource queries, but they perform poorly on the navigational class.

A recent work studied the linguistic difference between search queries and text documents,<sup>4</sup> discovering that approximately 70 percent of query terms are nouns and proper nouns, whereas adjectives are used around 7 percent of the time, and URLs 6 percent. As for documents, almost each sentence contained at least one verb. Because this poses a great challenge to conventional natural language-processing techniques, new ad hoc algorithms have been designed for dealing with search queries to assist in detecting user intention by using named-entity recognition techniques.<sup>5-9</sup>

## References

1. A. Broder, “A Taxonomy of Web Search,” *SIGIR Forum*, vol. 36, no. 2, 2002, pp. 3–10.
2. B.J. Jansen and D.L. Booth, “Classifying Web Queries by Topic and User Intent,” *Proc. ACM Conf. Human-Computer Interaction*, 2010, pp. 4285–4289.
3. I. Hernández et al., “A Simple Model for Classifying Web Queries by User Intent,” *Proc. 2nd Spanish Conf. Information Retrieval*, 2012, pp. 235–240.
4. C. Barr, R. Jones, and M. Regelson, “The Linguistic Structure of English Web-Search Queries,” *Proc. Conf. Empirical Methods in Natural Language Processing*, 2008, pp. 1021–1030.
5. A. Alasiry, M. Levene, and A. Pouloussis, “Extraction and Evaluation of Candidate Named Entities in Search Engine Queries,” *Web Information Systems Eng.*, LNCS 7651, Springer, 2012, pp. 483–496.
6. J. Du et al., “Using Search Session Context for Named Entity Recognition in Query,” *Proc. 33rd Int’l ACM Sigir Conf. Research and Development in Information Retrieval*, 2010.
7. J. Guo et al., “Named Entity Recognition in Query,” *Proc. 32nd Int’l ACM Sigir Conf. Research and Development in Information Retrieval*, 2009, pp. 267–274.
8. W. Ting-Xuan and L. Wen-Hsiang, “Identifying Popular Search Goals behind Search Queries to Improve Web Search Ranking,” *Proc. 7th Asia Conf. Information Retrieval Technology*, 2011, pp. 250–262.
9. A. Figueroa and G. Neumann, “Exploiting User Search Sessions for the Semantic Categorization of Question-Like Informational Search Queries,” *Proc. Int’l Joint Conf. Natural language Processing*, 2013, pp. 902–906.

a large portion of new queries by using simple heuristic patterns.

Thus, automatically detecting a user’s intention when searching is at the core of successful information retrieval systems on the Web. This task can usually be seen as a supervised learning problem (that is, classification) in which word-based learning algorithms (that is, classifiers) search through a hypothesis space to find a suitable hypothesis that will make good

predictions for an intention-detection problem.<sup>1,2</sup> Even if the hypothesis space contains hypotheses that are well-suited for a detection task, it might be difficult to find a good one.

To address similar tasks, methods ensembling multiple classifiers have caught the attention of the research community in the last 10 years.<sup>3</sup> Several strategies have been designed for tackling distinct problems, for instance, for semantic classification of search queries.<sup>4,5</sup>

In this article, we propose a novel approach based on an ensemble of classifiers. Unlike previous approaches, our research takes advantage of a specific type of ensembles via classifier selection to improve the recognition of the user's intent behind search queries. The model combines syntactic and semantic features so as to effectively detect a user's intention using different ensembling techniques for detecting intentions.<sup>6,7</sup>

### Ensembling Classifiers for Detecting User Intentions

Classification or supervised learning is a machine learning task of inferring a function from labeled training data. The training data consist of a set of labeled examples, which are pairs consisting of an input object and a desired output value. Ensemble learning refers to a collection of classification methods that learn a target function by training a number of single classifiers and combining their predictions. The principle is that a committee decision, with individual predictions combined appropriately, should have better overall accuracy on average than any individual committee member. For many tasks, ensemble models often attain higher accuracy than single models, because a more reliable function-sample mapping can be obtained by combining multiple experts' output.

Accordingly, this work addresses automatic recognition of user's intentions behind Web queries by extending and ensembling current classification models so as to improve the search experience. Instead of focusing on single classifiers for detecting different types of intentions,<sup>8</sup> in this research we explore ensembles of single classifiers.

To deal with the task of user intention detection using multiple types of queries, our fusion approach considers supervised, single classifiers that can easily cope with multiclass problems:

- *Multiclass support vector machines* (SVMs) are kernel-based supervised learning models with associated learning algorithms that classify data into several categories (see [http://svmlight.joachims.org/svm\\_multiclass.html](http://svmlight.joachims.org/svm_multiclass.html)).<sup>9</sup>
- The *Naive Bayes classifier* is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions between the features (see <http://mallet.cs.umass.edu>).

- A *maximum entropy* (MaxEnt) classifier is a probabilistic classifier based on the principle of maximum entropy, which assumes the features to be conditionally independent (see [www.cs.cmu.edu/~abberger/maxent.html](http://www.cs.cmu.edu/~abberger/maxent.html)). MaxEnt selects the model fitting the training data that has the largest entropy.
- A *multilayer perceptron* (MLP) is a feed-forward artificial neural network model that maps sets of input data onto a set of appropriate outputs (see [www.cimne.com/flood/](http://www.cimne.com/flood/)). An MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. MLP uses a supervised learning technique called *backpropagation* for training the network.

To train these learners (classifiers), we captured and annotated a corpus of Web queries from the AOL query collection. We then constructed three types of ensembles' classifier using several specific-purpose features:

- A *general ensemble* combines the four single classifiers to produce a final (intent) class label. Each classifier is general so that it has access to the entire query set when building its classification model.
- A *syntax-oriented ensemble* uses a syntactic-based taxonomy (that is, a classification of things or concepts, including the principles that underlie such classification) of search queries<sup>10</sup> by splitting each general, single classifier into four single classifiers that focus on a specific query type.
- A *length-oriented ensemble* creates different classifiers targeted at Web queries of distinct lengths. Unlike the previous ensembles, each single classifier is split into single classifiers based on the number of terms in the search query, so five groups were generated. Accordingly, each single classifier has access only to the respective query group when constructing its classification model.

### Corpus Acquisition

To prepare a dataset for detecting a user's intention, we used the AOL Web query corpus (see <http://gregsadetky.com/aol-data/>), which contains 21 million query instances submitted by approximately 650,000 search users. Each instance contains a user identification, time stamp, query

string, rank, and URL of the results clicked by the user. The overall collection consists of approximately 10 million distinct, lowercased queries, where 4,811,638 elements are linked with at least one clicked URL. These queries were then manually annotated. We first extracted a sample of 30,000 random queries from the remaining 3,788,459 unlabeled items, which 10 humans annotated. We provided each annotator with a set of 3,000 distinct queries and the description of each intention class so as to reduce ambiguity when tagging. As a result, manually annotated categories included 23,736 informational, 4,585 navigational, and 1,679 resource queries.

Because each annotator was assigned a different set of queries, we approximated the disagreement rate by inspecting the annotations for 100 random instances tagged by each annotator. The lowest and the highest disagreement were 2 and 15 percent, respectively, with an average of 8 percent (with a standard deviation of 4.37).

Despite the preliminary annotation results, manual tagging is a time-demanding task, and might be biased by human criteria. In addition, researchers have observed that human intentions aren't that ambiguous, so there's no need for too many annotators. Hence, we combined manual annotation with rule-based automatic annotation. For this, we defined some rules to include common words that unambiguously imply a particular class of query intention.<sup>11</sup>

By using these rules, only 21.26 percent of the queries were automatically tagged in our research, whereas other elements required manual annotations. From these tagged samples, we manually inspected 300 randomly selected instances (with an error rate of 7.33 percent). Overall, 1,023,179 items were automatically extracted (793,314 navigational, 122,805 resource, and 107,060 informational). The annotated datasets can be obtained from [www.inf.udec.cl/~atkinson/AnnotatedCorpus.rar](http://www.inf.udec.cl/~atkinson/AnnotatedCorpus.rar).

## Ensembling Single Learners

In our approach, we use two ensembling strategies:

- *Borda count method*. This is a single-winner election method, in which voters rank options or candidates in order of preference. The Borda count determines the outcome of a debate by giving each candidate, for each ballot, a number of points corresponding to the number of candidates ranked lower. Once

all votes have been counted, the candidate with the most points is the winner.

- *MaxEnt ensemble*. MaxEnt ensembling models possess several desirable features such as flexibility in adding new features, scalable training, easy parameter estimation, and minimal assumptions about the posteriors. In our approach, MaxEnt models are trained with two kinds of features: the user's intention returned at each ranking position by each learner, and binary features indicating whether two or three classifiers rank the same intention in the same position. We then apply a greedy algorithm to select the best features for this type of ensemble.

## Features for Each Single Learner

Unlike other approaches,<sup>8</sup> we captured several features so as to take advantage of each single learner. Overall, we identified five groups of features: term-level features, caseless models, named entities in queries, Barr's taxonomy, and query expansion.

**Term-level features.** These features include elements such as words and the query length.

**Caseless models.** These are a set of fine-grained features extracted from caseless corpora by conducting some natural language processing (NLP)<sup>12</sup> tasks, including the following:

- *Named-entity recognition (NER)*. This task identifies and classifies atomic elements (named entities) in text into predefined categories, such as the names of persons, organizations, locations, and so on.
- *Dependency parsing*. In NLP, parsing or syntactic analysis is the task of analyzing a sentence of words into its structure, resulting in a parse tree showing their syntactic relation to each other, which might also contain semantic and other information. A typical structure is based on constituents (that is, noun phrases linked to verb phrases, and so on) or dependency relations. A dependency relation views the (finite) verb as the structural center of all clause structure. All other syntactic units (for example, words) are either directly or indirectly dependent on the verb. Thus, structure is determined by the relation between a word (a head) and its dependents.

By using state-of-the-art NLP methods and publically available tools (see <http://nlp.stanford.edu/software>), extracted features included named entities representing organizations, persons, and locations; and dependency relations extracted from the queries' dependency trees using the Stanford Dependency Parser. Obtained dependency information (dependency paths) included the following:

- the number of dependency relations (out of 109 distinct values from 6,000 queries),
- the total number of dependency relations,
- lexical relationships such as full (typed) relations (that is, *prep\_in* : falling → love) or partial (typed) relations (that is, *prep\_in* : falling), and
- root model features such as the value and position of the root node.

**Named entities in queries.** We added Boolean features to indicate the presence or absence of 20 different categories of entities distinguished by a NER such as brand name, business, disease and condition, dish, food, place name and product, and so on.

**Barr's Taxonomy.** This group query's Boolean features representing syntactical information obtained from a specific-purpose taxonomy.<sup>10</sup> These identify queries that are noun phrases, questions, URLs, and verb phrases.

**Query expansion.** This feature reformulates a query to improve document retrieval performance, and involves evaluating a user's input and expanding the search query to match additional documents. To carry out this expansion, we exploited several sources of semantic information:

- *WordNet*. This lexical database (see <http://wordnet.princeton.edu>) groups words into sets of semantic synonym relationships called synsets, recording a number of relations among these synonym sets or their members. By using WordNet, we found 26 distinct semantic relations across our collection including hypernyms (for example, *cover* → *conceal*) and meronyms (for example, *motorcycle* → *kick starter*).
- *Wikipedia-based features*. We included six Boolean features to indicate whether a query

was expanded with words contained in five different sections of Wikipedia: abstracts, first paragraphs, categories, infobox types, and sense discriminators. To look-up articles, we look for a case insensitive match between the title and the search query, and include a Freebase category related to that matched article. To map Wikipedia articles into Freebase categories, we used the WEX (see <http://wiki.freebase.com/wiki/WEX>).

Features for each single classifier are selected by using a greedy search algorithm, which follows the problem solving heuristic of making the locally optimal choice at each stage so as to hopefully find a global optimum (or a global approximate optimal solution in a reasonable time). The algorithm starts with an empty bag of features and after each iteration adds the one that performs the best. To determine this feature, this search method tests each nonselected features together with all the features in the bag. The algorithm stops when there is no nonselected features that improve the performance.

## Experiments and Results

To assess the performance of our multiclass ensembling method, we conducted several experiments using different ensembling configurations using the mean reciprocal rank (MRR), which is a statistic metric for evaluating any process that produces a list of possible responses to a sample of queries, ordered by probability of correctness. Note that this is preferable to  $P@n$  (that is, precision at  $n$ ) as MRR provides much information on the classification errors. The MRR is the average of the reciprocal ranks of results for a sample of queries  $Q$ :

$$\frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i}$$

Furthermore, for benchmark purposes, we considered two baselines:

- We built a *centroid vector* for each user's intention. Each testing sample was then assigned the class label corresponding to the best scoring centroid vector. This baseline has an MRR of 0.8376.
- We trained *single classifiers* with words as features. The highest MRR score was 0.8675 using MLP, which we set as the baseline.



We trained and tested classifiers for both baselines, and ensembled them by using the Borda count and MaxEnt methods.

To assess our different ensembles (that is, general-purpose classifiers, syntax-oriented classifiers, and length-oriented classifiers), we performed a  $k$  – fold ( $k = 10$ ) cross-validation. Here the original sample is randomly partitioned into  $k$  equally sized subsamples, where one is retained as the test data, and the remaining  $k - 1$  subsamples are used for training. This process is then repeated  $k$  times (the folds), with each of the  $k$  subsamples used exactly once as the test data.

Table 1 shows an overview of the three configurations for our experiments. In the first place, all ensembles outperformed the use of single models, independently on exploiting an unsupervised approach (Borda counts) or a supervised approach (MaxEnt). However, the supervised approach finished with better results than the unsupervised counterpart. Accordingly, Borda counts might be a cost-efficient solution to improve performance. On the other hand, the fact that configuration II and III outperformed configuration I indicates that the classifier selection is a promising contribution to this task. In words, splitting the search query space according to some intrinsic features (for example, their length or some syntactic pattern) and building classifiers specialized in each of those features improves the overall performance. In our case, computing the number of tokens/terms within a search query is efficient because it's computationally inexpensive.

The improvement of the method using classifier selection might be because the feature optimization algorithm filters those elements that are more suitable to each region. Furthermore, even though some features might be included in several specialized classifiers, the distribution of their values might radically differ from one region to the other. As a natural consequence, focused classifiers can capture these differences across intentions more effectively. Thus, improving the ranking of intention is a key factor to tailor results that fit the display into small modern devices such as tablets and mobile phones.

### Configuration I: General-Purpose Ensemble

Each general-purpose single classifier outperformed both baselines. The worst single classifier (that is, MLP) obtained an MRR of 0.8988, which is 3.61 percent better than the other

Table 1. Overview of performance for the proposed ensembles.\*

| Configuration | Borda count | MaxEnt ensemble |
|---------------|-------------|-----------------|
| I             | 0.9128      | <b>0.9143</b>   |
| II            | 0.9124      | <b>0.9156</b>   |
| III           | 0.9163      | <b>0.9211</b>   |

\* Bold indicates the best performance achieved in terms of mean reciprocal rank (MRR).

classifiers using words as features (MRR = 0.8675). Overall, the best classifier (MaxEnt) outperformed the worst one by 0.76 percent (MRR = 0.9101), suggesting that the set of previously defined features is useful for automatically detecting the user's intention. Possibly this is because beyond bag-of-words, no other features were incorporated into the four classifiers. However, four features are used in three of the classifiers: number of relations (extracted from the dependency tree using a parser), file names identified by the NER task, query expansion terms contained in the first paragraph of Wikipedia articles, and first-level categories provided by Wex. The most discriminative cues were indeed extracted from Wikipedia and Wex: The people category was linked 68 percent with navigational queries; whereas this category only linked 32 percent with informational intentions. Furthermore, the music class was linked 80 percent with navigational intentions; whereas it linked 20 percent with informational intentions. In general, getting a first-level Wex category for the query is associated 76 and 24 percent with navigational and informational queries, respectively. On the other hand, file names were related 83 percent to resource queries and 17 percent to informational intentions.

Because both types of ensembles might improve the performance compared with the best single classifier (MaxEnt) with MRR = 0.9101: Borda counts by 0.30 percent and MaxEnt ensemble by 0.46 percent. Hence, further experiments assessing other types of classifier ensembles become promising. Results also show that a supervised approach such as the MaxEnt ensemble can bring further benefits, as key features are associated with top-ranked intentions of each classifier. Experiments also suggest that an SVM outperforms the other classifiers when dealing with informational queries (MRR = 0.9852), whereas an MLP performs better on the other two intentions (MRR = 0.9037 for navigational and

Table 2. Ensembling syntax-oriented single learners (configuration II).

| Single learner          | No. of samples | Multilayer perceptron (MLP) | MaxEnt | Bayes  | Support vector machine (SVM) | Ensemble     |               |
|-------------------------|----------------|-----------------------------|--------|--------|------------------------------|--------------|---------------|
|                         |                |                             |        |        |                              | Borda counts | MaxEnt        |
| Question                | 2,943          | 0.9929                      | 0.9898 | 0.9911 | 0.9893                       | 0.9919       | <b>0.9949</b> |
| URL                     | 10,119         | 0.9965                      | 0.9966 | 0.9968 | 0.9967                       | 0.9967       | <b>0.9973</b> |
| Noun phrase             | 24,590         | 0.8553                      | 0.8760 | 0.8717 | 0.8270                       | 0.8766       | <b>0.8806</b> |
| Others                  | 22,348         | 0.8784                      | 0.8991 | 0.8921 | 0.8818                       | 0.9031       | <b>0.9066</b> |
| Syntax-oriented learner | 60,000         | 0.8945                      | 0.9105 | 0.9063 | 0.8840                       | 0.9124       | <b>0.9156</b> |

MRR = 0.8583 for resources queries). Unlike previous work,<sup>8</sup> the multilayer nature of perceptron classification models are promising when compared with Bayes classifiers, as they can capture more complex relationships between queries and target classes. Nevertheless, the performance of the SVM significantly drops when dealing with resource (MRR = 0.6226) and navigational (MRR = 0.8395) queries, so the performance of the MLP drops on the informational type as it achieves an MRR of 0.9019 (8.5 percent lower than for SVM), showing a significant variance across the three intentions for these two learners.

Experiments also indicated that Bayes and MaxEnt get higher MRR scores than the other classifiers, improving the rank. Indeed, MaxEnt ensemble didn't outperform the SVM on the informational class, nor the MLP model on the other two intentions. However, MaxEnt got the best overall MRR score, which might be because weighting the outcomes of multiple classifiers reduces large variances across distinct intentions as compared to single learners.

### Configuration II: Syntax-Oriented Ensemble

For this experiment, we split the general classifier into four single classifiers based on a syntactic taxonomy,<sup>10</sup> where a single classifier is a classifier focused on a specific query type (that is, a smaller group of queries showing a specific pattern).

As Table 2 shows, using smaller units assisted the MaxEnt and Bayes classifier to improve classification accuracy with a slightly higher MRR score than previous settings (0.9101 to 0.9105 and 0.9056 to 0.9063, respectively). On the other side, for MLP and SVM classifiers, the MRR performance decreased from 0.8988 to 0.8945 and 0.8853 to 0.8840, respectively, suggesting that using all-data encompassing classifiers is a much more cost-efficient than grouping the same classifier into syntactically targeted units.

In terms of syntax-oriented classifiers, MLP, Naive Bayes, and MaxEnt classifiers got the

best performance for the question type and for the URL and noun phrase groups, respectively. However, all syntax-oriented single classifiers achieved a high performance for URL and question groups, because these groups are biased toward a particular intent. Furthermore, most of the instances contained in the URL group are navigational (99.37 percent); whereas they're informational queries in the question group (97.86 percent). The distribution of intentions across the noun phrase group is similar to that found across the 60,000 queries: 43.31 percent for informational, 48.68 percent for navigational, and 8 percent for resources. However, for the others class, 59.71 percent are informational, whereas 25 percent are navigational, showing that the intentions of noun phrases are the most difficult to detect, followed by the others group. Nevertheless, an ensemble of single classifiers improved the performance of the best single classifier (MaxEnt) on the noun phrase group from 0.8760 to 0.8806 (0.53 percent). For query intentions, the MLP classifier outperformed the other three classifiers when coping with the navigational (0.9032) and resource (0.85) queries, whereas the Bayes classifier was the best on the informational class with an MRR score of 0.9477, mainly due to the few features incorporated into more than one of the single learners. In particular, for MLP, MaxEnt, Bayes, and SVM only 2, 8, 5, and 2 features, respectively, were considered into more than a single classifier, where duplicated features were often due to named entities extracted via NER tasks.

### Configuration III: Length-Oriented Ensemble

For length-oriented single classifiers, the performance was better than for configurations I and II, as Table 3 shows. The SVM's performance increased by 2.39 percent (with configuration I using two-tailed  $t$ -test = 3.92,  $n$  = 20,  $\alpha$  = 0.01, and  $p$  < 0.001), whereas the Naive Bayes



**Table 3. Mean reciprocal rank scores for each single learner, length-oriented classifier, and both ensembles (configuration III).**

| Length                  | No. of samples | MLP    | MaxEnt | Bayes  | SVM    | Ensemble     |               |
|-------------------------|----------------|--------|--------|--------|--------|--------------|---------------|
|                         |                |        |        |        |        | Borda counts | MaxEnt        |
| 1                       | 15,260         | 0.9680 | 0.9681 | 0.9683 | 0.9681 | 0.9683       | <b>0.9687</b> |
| 2                       | 12,868         | 0.8301 | 0.8414 | 0.8444 | 0.8311 | 0.8467       | <b>0.8554</b> |
| 3                       | 11,874         | 0.8877 | 0.9098 | 0.9066 | 0.9005 | 0.9102       | <b>0.9152</b> |
| 4                       | 8,606          | 0.8987 | 0.9104 | 0.9091 | 0.9073 | 0.9138       | <b>0.9203</b> |
| 5+                      | 11,392         | 0.9247 | 0.9243 | 0.9277 | 0.9148 | 0.9338       | <b>0.9380</b> |
| Length-oriented learner | 60,000         | 0.9044 | 0.9128 | 0.9133 | 0.9065 | 0.9163       | <b>0.9211</b> |

classifier ( $p < 0.001$ ), showed a slight growth by 0.30 percent. The Borda counts ensemble also improved the MRR score from 0.9128 (configuration I) to 0.9163. Furthermore, both ensembles outperformed all four length-oriented single learners, where MaxEnt ensemble improved by 0.74 percent ( $t$ -test = 3.675,  $n = 20$ ,  $\alpha = 0.01$ , and  $p < 0.001$ ), suggesting that a length-oriented ensemble significantly performs better than a syntax-oriented ensemble.

For length-oriented groups, the Bayes classifier had the best performance for queries composed of 1, 2, and 5+ terms, whereas MaxEnt did well for queries containing 2 and 3 terms. For each group, Borda counts outperformed the corresponding single learner, and MaxEnt ensemble outperformed Borda counts. Note that the increase of MRR scores is proportional to the number of terms, suggesting that the more context provided by the query, the higher the performance. Looking at the number of query expansion features selected for each length – length 2 single learners chose these 23 times, whereas single classifiers targeting longer queries chose these 10 to 11 times – we see that the lack of context radically affect two-term queries, and consequently, query expansion features become key items for resolving the intention for this query type.

Experiments showed that the most significant query expansion features included Wikipedia sense discriminators and the Wex categories, indicating useful query expansion features and that only a few terms (most likely one term) signal the query topic.

Single classifiers aimed at two-term queries rely on a large number of class features, as they provide useful context and narrow coverage. For instance, Wex categories were only found for 8.57 percent of the elements of this group. As for one-term queries, they're 93.70 percent

of the time navigational, which makes it easier to guess their intent, whereas two-term queries are 37.61 percent and 56.50 percent informational and navigational queries, respectively.

For query intentions, the SVM got the best results when dealing with informational intentions (0.9521). Bayes did well for navigational (0.9063), and MLP did well for resources (0.8630). Overall, features extracted from dependency trees and NER tasks were significant for building effective intention classifiers. It's worth noting that three out of the four single classifiers used the pair partial relations and NER domains when resolving the queries' intention containing 4 and 5+ terms. Furthermore, Wex first-level categories and brand names were significant to three out of the four single classifiers dealing with queries composed of two terms.


It was difficult to infer intentions for Web queries using short-length queries made of two terms, especially noun phrase queries (only 39.78 percent of the noun phrase queries had two terms, and 76.03 percent of two-term queries are noun phrases). Thus, the MRR score for the MaxEnt ensemble for this intersection was only 0.8525, whereas it was 0.8647 for the remaining 23.97 percent.

Thus, our results for intent classification indicate that length-based ensembles are the best choice, because the classifier selector only needs the token count to select the right set of classifiers, while achieving the best overall performance.

**T**his work has proposed a new multiclass ensembling strategy for automatically recognizing a user's intentions behind Web queries. Our approach combines stochastic machine learning techniques and two ensemble methods to take advantages of multiple features extracted from different sources, including knowledge bases, the query, and other electronically available resources.

Experiments using our model assess different configurations for features, ensembling methods and classifiers showing that combining classifiers' outcomes assists in improving the quality of the user's intentions measured as position in a ranking of the best candidate intentions. Configurations of ensembles were composed of targeted classifiers; that is, single classifiers aimed at specific lengths and syntactic patterns, indicating that designing ensembles with focused classifiers improved the ranking of user's intentions, as compared with single classifier approaches.

Incorporating a classifier-selection task performed well when compared with other classification methods, which might be because the feature optimization algorithm is capable of filtering those elements that are more suitable to each region. Even though some features might be included in several specialized classifiers, the distribution of its values might radically differ from one region to the other. As a natural consequence, focused classifiers can capture these differences across intentions more effectively. In real-life applications, a key factor is to tailor search results that fit the display in small modern devices such as tablets and mobile phones.

As future work, given the high cost of annotating search strings by humans and the availability of large amounts of unlabeled data, we envision the use of semi-supervised or multi-view learning to enhance the classification rate, especially of two term queries. In terms of application, our results can assist any information retrieval system that indexes Web-like documents. 

### Acknowledgments

This research was partially supported by FONDECYT (Chile) research project 11130094 ("Bridging the Gap between Askers and Answers in Community Question Answering Services") granted to Alejandro Figueroa, and by FONDECYT (Chile) research project 1130035 ("An Evolutionary Computation Approach to Natural language Chunking for Biological Text Mining Applications") granted to John Atkinson.

### References

1. I.-H. Kang and G.-C. Kim, "Query Type Classification for Web Document Retrieval," *Proc. 26th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval*, 2003, pp. 64–71.
2. U. Lee, Z. Liu, and J. Cho, "Automatic Identification of User Goals in Web Search," *Proc. 14th Int'l Conf. World Wide Web*, 2005, pp. 391–400.

3. L.I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, John Wiley & Sons, 2004.
4. D. Shen et al., "Q 2 C @ UST: Our Winning Solution to Query Classification in KDDCUP 2005," *ACM SIGKDD Explorations Newsletter*, vol. 7, no. 2, 2005, pp. 100–110.
5. Y. Li, Z. Zheng, and H. (K.) Dai, "KDD Cup-2005 Report: Facing a Great Challenge," *ACM SIGKDD Explorations Newsletter*, vol. 7, no. 2, 2005, pp. 91–99.
6. S.M. Beitzel et al., "Automatic Classification of Web Queries Using Very Large Unlabeled Query Logs," *ACM Trans. Information Systems*, vol. 25, no. 2, 2007.
7. D. Shen et al., "Query Enrichment for Web-Query Classification," *ACM Trans. Information Systems*, vol. 24, no. 3, 2006, pp. 320–352.
8. I. Hernández et al., "A Simple Model for Classifying Web Queries by User Intent," *Proc. 2nd Spanish Conf. Information Retrieval (CERI)*, 2012, pp. 235–240.
9. K. Crammer and Y. Singer, "On the Algorithmic Implementation of Multi-Class SVMs," *Proc. J. Machine Learning Research*, vol. 2, 2001, pp. 265–292; <http://jmlr.csail.mit.edu/papers/volume2/crammer01a/crammer01a.pdf>.
10. C. Barr, R. Jones, and M. Regelson, "The Linguistic Structure of English Web-Search Queries," *Proc. Conf. Empirical Methods in Natural Language Processing*, 2008, pp. 1021–1030.
11. B.J. Jansen and D.L. Booth, "Classifying Web Queries by Topic and User Intent," *Proc. ACM Conf. Human-Computer Interaction*, 2010, pp. 4285–4289.
12. D. Jurafsky and J.H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 2nd ed., Prentice Hall, 2008.

**Alejandro Figueroa** is a researcher at Yahoo Research, Universidad Andres Bello, and Universidad Diego Portales, Santiago, Chile. His research interests include question-answering systems, natural language processing, machine learning, and information retrieval. Figueroa has a PhD in computational linguistics from Universitaet des Saarlandes, Saarbruecken, Germany. Contact him at [afigueroa@yahoo-inc.com](mailto:afigueroa@yahoo-inc.com).

**John Atkinson** is a full professor in the Department of Computer Sciences, Universidad de Concepción, Concepción, Chile. His research interests include basic and applied research in text mining, natural language processing, artificial intelligence, and machine learning. Atkinson has a PhD in artificial intelligence from the University of Edinburgh, Scotland. He's a member of the American Association for Artificial Intelligence and the IEEE Computer Society, and a senior member of ACM. Contact him at [atkinson@inf.udec.cl](mailto:atkinson@inf.udec.cl).