What Makes a Query Semantically Hard?

Guglielmo Faggioli*
guglielmo.faggioli@phd.unipd.it
University of Padova
Italy

Stefano Marchesin*
stefano.marchesin@unipd.it
University of Padova
Italy

ABSTRACT

Traditional Information Retrieval (IR) models, also known as lexical models, are hindered by the semantic gap, which refers to the mismatch between different representations of the same underlying concept. To address this gap, semantic models have been developed. Semantic and lexical models exploit complementary signals that are best suited for different types of queries. For this reason, these model categories should not be used interchangeably, but should rather be properly alternated depending on the query. Therefore, it is important to identify queries where the semantic gap is prominent and thus semantic models prove effective. In this work, we quantify the impact of using semantic or lexical models on different queries, and we show that the interaction between queries and model categories is large. Then, we propose a labeling strategy to classify queries into semantically hard or easy, and we deploy a prototype classifier to discriminate between them.

KEYWORDS

semantic models, semantically hard query, model selection

ACM Reference Format:
Guglielmo Faggioli and Stefano Marchesin. 2021. What Makes a Query Semantically Hard?. In *Design of Experimental Search & Information REtrieval Systems, September 15–18, 2021, Padua, Italy*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

The semantic gap is a long-standing problem in Information Retrieval (IR) that refers to the difference between the machine-level description of document and query contents and the human-level interpretation of their meanings [21]. In other words, it represents the mismatch between users' queries and the way retrieval models understand such queries [41].

The semantic gap affects any domain, but it is prominent in medical search [9, 20, 21]. Within biomedical literature, the large presence of (quasi-)synonymous and polysemous terms – along with the use of acronyms and terminological variants – represents a critical challenge for retrieval models. In this regard, a query containing the word "tumor" might not be effectively answered if the retrieval model does not identify the synonymy relationship between "tumor" and, for example, "neoplasm". Besides, given a query containing the term "cold", a retrieval model might retrieve erroneous documents if it does not distinguish between the different meanings the term "cold" assumes depending on the context, such as "common cold", "cold temperature", or even "Chronic Obstructive Lung Disease". These queries are known as *semantically hard* queries [1].

Traditional IR models, which are known as lexical models as they compute the relevance score using heuristics defined over the lexical overlap between queries and documents, fail to effectively address semantically hard queries. Semantic models were thus introduced to bridge the semantic gap [23] and to overcome the limitations of lexical models. However, semantic models have been shown to provide complementary signals to lexical models that prove effective for semantically hard queries, but less for other queries [26]. Thus, it becomes necessary to identify what category of models – between lexical and semantic – best suits a user query given the document collection at hand. In other words, we need to understand what are the inherent features of query and documents that make lexical or semantic models more effective.

To this end, we address the following research questions:

RQ1 How and to what extent does the semantic gap impact query performance?
RQ2 What features determine the prominence of the semantic gap within queries?

For **RQ1**, we investigate and compare the impact of lexical and semantic models on different topics. How large is the interaction between topics and model categories? To what extent does this interaction reflect in the different topic formulations (i.e., queries)? For **RQ2**, we explore a different set of well-known features that relate to lexical and semantic models. In particular, we seek to understand whether pre-retrieval features – based on corpus statistics or synonymy/polysemy aspects – can be used to categorize queries as semantically easy or hard. In other words, how effective are well-known pre-retrieval features for category selection?

To address the research questions, we first perform statistical analyses quantifying the interaction between topics, queries, and lexical and semantic categories using ANalysis Of VAriance (ANOVA) [31]. Based on the outcomes of the statistical analyses, we propose a labeling strategy to categorize queries into semantically easy or hard. The labeled queries are used to train a category selector. The selector serves as a proxy to evaluate the effectiveness of the considered pre-retrieval features in determining the prominence of the semantic gap within queries.

We conduct an experimental evaluation on two test collections for ad hoc medical retrieval: OHSUMED [18] and TREC-COVID

*Authors contributed equally to this work.

Desires '21, September 15–18, 2021, Padua, Italy

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

(Round 1) [36]. For lexical models, we adopt standard state-of-the-art retrieval models. Regarding semantic models, we focus on first-stage semantic models, which are best suited to tackle the semantic gap [35]. In particular, we consider unsupervised first-stage semantic models, which have shown to be competitive with lexical models in medical collections [1]. Besides, unsupervised semantic models rely on textual signals only – and not on relevance signals – thus allowing us to focus exclusively on semantic and lexical features.

The results of the experimental evaluation show that topics, queries, and model categories strongly interact to determine retrieval effectiveness. This evidence further highlights the need to adopt the proper model category to improve retrieval performance. Therefore, identifying the right features to distinguish between semantically easy or hard queries becomes necessary in domains where the semantic gap is prominent – and this work poses the cornerstone towards this direction.

The rest of the paper is organized as follows: Section 2 reports related work; Section 3 presents the experimental analysis; and Section 4 concludes the paper and outlines the future directions.

2 RELATED WORK

The problem addressed in this work relates to two topics in IR: Model Selection and Query Performance Prediction (QPP). Below, we review prominent approaches in these areas and we highlight differences with our work.

One of the first approaches to model selection in IR was developed by He and Ounis [16], who proposed a query-based pre-retrieval approach. In [16], the authors cluster queries according to pre-retrieval features and link the best performing model to each cluster. Then, given a new query, they assign it to the closest cluster and use the model associated to that cluster to perform retrieval. Balasubramanian and Allan [5] proposed a learning approach for query-dependent model selection. The selection framework relies on rank-time features – available to retrieval models during ranking – to select between two models. Model selection approaches based on rank-time features have been further explored by Balasubramanian in [4]. Beyond model selection, Levi et al. [22] addressed the problem of selective cluster retrieval [14, 24, 34], where the objective is to decide, on a per-query basis, whether to apply cluster-based retrieval or standard document retrieval. In [22], the authors proposed different sets of features based on cluster-based rankers, query performance predictors, and cluster properties. The different sets of features are then used to decide between cluster-based and standard document retrieval.

Compared to the approaches reviewed for model selection, in this work we want to understand whether queries can be categorized as semantically easy or hard. In other words, we want to determine which models category between lexical and semantic is best suited on a per query basis. In this sense, our work shows similarities with that of Levi et al. [22], where the objective is to select the most effective approach between cluster-based and document-based retrieval given the query. However, we refrain from using rank-time or post-retrieval features in our analyses, as we want to keep the approach model-agnostic – and thus less dependent on the specific sets of considered retrieval models.

QPP techniques are traditionally divided into pre-retrieval and post-retrieval. Pre-retrieval techniques [8, 15, 17, 28, 42] exploit the distribution of the query terms within the collection, providing coarse-grained information on the expected performance of a given query. On the other hand, post-retrieval techniques [3, 30, 33] leverage the information on the retrieval scores assigned by the retrieval model. Such techniques tend to perform better compared to pre-retrieval QPP [11], but are dependent on the considered models.

The typical task for a QPP model is ranking queries based on their expected performance [11]. Thus, QPP techniques cannot be directly applied to category selection. Nevertheless, the signals provided by QPP models can be used as input features for such task. In this work, we want to identify a query as semantically easy or hard regardless of the retrieval model considered. Thus, we focus on pre-retrieval approaches and we adopt two types of features in our analyses: lexical- and semantic-oriented features. Regarding lexical-oriented features, we consider features proposed by He and Ounis [17] and by Zhao et al. [42]. He and Ounis [17] explore the possibility to use the distribution of the Inverse Document Frequency (IDF) over query terms to determine the ability of lexical models to retrieve relevant documents. Similarly, Zhao et al. [42] propose a re-weighting schema based on IDF, called Similarity between Collection and Query (SCQ). As for semantic-oriented features, we adopt features similar to those proposed by Mothe and Tanguy [28], who consider linguistic aspects – such as synonymy and polysemy – linked to the query terms. Compared to [28], however, we consider signals from both the query and its interaction with documents.

3 EXPERIMENTAL ANALYSIS

We consider two collections in the following analyses: OHSUMED [18] and TREC-COVID (Round 1) [36]. OHSUMED contains 349K documents and 63 topics. Topics in OHSUMED have two fields: *title* and *description*. We use *description* as topic formulation since the *title* field poorly describes the underlying information need. TREC-COVID (Round 1) has 30 topics and relies on the CORD-19 corpus [39], which includes around 51K papers. Each topic in TREC-COVID has three fields: a short keyword *query*, a *description*, and a *narrative*. In our experiments, we consider each field as a different formulation of the topic. We also include the *concatenation* of the keyword query and the description. Thus, the total number of queries we consider for TREC-COVID is equal to 120.

Regarding lexical and semantic models, we consider five different models for each category. The lexical models used are: TF-IDF [7]; BM25 [29]; Query Likelihood Model with Dirichlet Smoothing (QLM) [40]; Divergence From Randomness (DFR) [2]; and Divergence From Independence (DFI) [19]. All lexical models perform stopwords removal and stemming. As for semantic models, we adopt: a Word2Vec [27] based approach where query and document representations are built by summing up the IDF-weighted representation of the words contained in them [25, 38]; the Neural Vector Space Model (NVSM) [35]; and three variants of the Semantic-Aware neural Framework for IR (SAFIR) [1]. The three variants of SAFIR are SAFIR_{sp}, which integrates both polysemy and synonymy, SAFIR_{p} which integrates polysemy but not synonymy,

Table 1: Mean Average Precision@1000 (MAP) of the models on OHSUMED and TREC-COVID collections. Models performance are comparable both within and across models categories.

Model	OHSUMED	TREC-COVID
	Lexical	
TF-IDF	0.524	0.362
BM25	0.620	0.488
QLM	0.577	0.434
DFR	0.641	0.496
DFI	0.592	0.467
	Semantic	
Word2Vec	0.568	0.482
NVSM	0.595	0.455
SAFIR$_s$	0.604	0.463
SAFIR$_p$	0.610	0.461
SAFIR$_{sp}$	0.612	0.466

and SAFIR$_s$ which integrates synonymy but not polysemy. All semantic models have been trained for 10 epochs with parameters set as in [1].

We evaluate models using Average Precision (AP) at cutoff 1000, obtaining an experimental Grid of Points (GoP) as defined in [12]. The performances of the retrieval models in terms of AP are reported in Table 1 for both OHSUMED and TREC-COVID collections.

3.1 RQ1: Topic and Category Interaction

Several works have shown that queries strongly interact with retrieval models in determining their performance [6, 13]. This means that two models might have similar average performance on a set of queries but, when looked at the query-level, their performance might vary greatly. A similar consideration also applies to lexical and semantic models. Some queries are best suited to semantic models, while some others to lexical ones [1, 26]. We are thus interested in quantifying such an effect. In other words, we want to evaluate the interaction between queries and model categories.

To determine whether the models category – that is, lexical or semantic – has a significant effect on performance, we conduct an ANOVA on the runs obtained with the considered retrieval models. ANOVA is a well-known statistical technique that allows identifying statistically significant differences among experimental conditions. Several works in IR applied ANOVA to determine the effect of different factors on the overall performance of an IR system [6, 10, 13, 37]. ANOVA models the explained variable, which in our case is AP, as a linear combination of the effect of each factor in the experimental setup, plus an error component. The error term accounts for the variance in the data unexplained by the model.

In our analyses we first consider the following model:

$$y_{ijk} = \mu_{...} + \tau_i + \gamma_j + \alpha_{k(j)} + \tau\gamma_{ij} + \varepsilon_{ijk}, \qquad \text{(MD1)}$$

where y_{ijk} is the performance (measured using AP) observed on the i-th topic using the k-th model of the j-th class; $\mu_{...}$ is the grand mean over all the data; τ_i is the effect of the i-th topic; γ_j is the effect of the j-th class; $\alpha_{k(j)}$ is the effect of the k-th model inside the j-th class; $\tau\gamma_{ij}$ is the interaction between the i-th topic and the j-th class and ε_{ijk} is the prediction error. Note that the *model* factor is nested inside the *category* one. In the above-mentioned ANOVA model, a IR model is meaningful only in relation to its category. In other words, since we cannot consider, for instance, BM25 inside the "semantic" category, nor we can consider NVSM in the "lexical" one, we define the model factor as nested inside the category, and thus each model contributes only to the variance of its category.

Table 2: ANOVA summary table on runs for the OHSUMED collection. Observe the large interaction between the topic factor and category factor. ω^2 for not significant factors is ill-defined and thus not reported.

Source	SS	DF	MS	F	p-value	$\hat{\omega}^2_{\langle fact \rangle}$
Topic	19.740	62	0.318	79.831	$< 1e-4$	0.886
Category	0.007	1	0.007	1.805	0.1797	—
Model(Category)	0.584	8	0.073	18.306	$< 1e-4$	0.180
Topic*Category	1.583	62	0.026	6.403	$< 1e-4$	0.347
Error	1.978	496	0.004			
Total	23.892	629				

For each ANOVA, we report the Sum of Squares (SS), the Degrees of Freedom (DF), the Mean Squares (MS), the F-statistic (F), the p-value and the Strength of Association (SOA), using the ω^2 indicator. The SOA indicates the impact of each factor on the variability of the data. Typically, a factor with $0.01 \leq \omega^2 < 0.06$ is considered small-sized, while $0.06 \leq \omega^2 < 0.14$ indicates a medium-size effect, and $\omega^2 \geq 0.14$ a large-size effect. Table 2 reports the results of the ANOVA on OHSUMED using the above-mentioned GoP of runs.

From the results in Table 2 we observe that the effect of the sole models category is not significant (p-value>0.05) – which means that lexical and semantic categories are not statistically significantly different. In other words, we cannot say that either lexical or semantic models perform best in absolute terms. Nevertheless, the interaction between topic and category is significant and the ω^2 value indicates a large effect. This means that the category significantly impacts on how good the results on a specific topic will be. Such a finding suggests that the semantic gap is an inherent property of the topics, less related to the specific retrieval models and more on their category. To further support this intuition, the interaction between the topic and the category is larger than the effect of the sole model. Thus, if we understand when a topic is lexical or semantic, we can achieve large performance improvements.

As for TREC-COVID, each topic is represented by four different formulations: the keyword *query*, the *description*, the *narrative* and the *concatenation* of query and description. Each formulation of a topic can only be used in relation to that topic and therefore the formulations have to be treated as a nested factor inside the topic. Therefore, we define a second ANOVA model, called MD2:

$$y_{iljk} = \mu_{...} + \tau_i + \phi_{l(i)} + \gamma_j + \alpha_{k(j)} + \tau\gamma_{ij} + \phi\gamma_{l(i)j} + \varepsilon_{ijlk}, \qquad \text{(MD2)}$$

which also includes $\phi_{l(i)}$, the effect of the l-th formulation, nested inside the i-th topic, and $\phi\gamma_{l(i)j}$, the interaction between the l-th formulation of the i-th topic with the j-th class. Table 3 summarizes the ANOVA results with MD2 on TREC-COVID.

Table 3: ANOVA summary table on runs for the TREC-COVID collection. Observe the high $\hat{\omega}^2$ effect for the interaction `topic*category` that shows the importance of selecting the proper model category for each topic.

Source	SS	DF	MS	F	p-value	$\hat{\omega}^2_{\langle fact \rangle}$
Topic	24.100	29	0.831	301.291	$< 1e-4$	0.879
Query(Topic)	15.568	90	0.173	62.712	$< 1e-4$	0.822
Category	0.074	1	0.074	26.732	$< 1e-4$	0.021
Model(Category)	1.470	8	0.184	66.628	$< 1e-4$	0.304
Topic*Category	2.200	29	0.076	27.506	$< 1e-4$	0.390
Query(Topic)*Category	1.060	90	0.012	4.270	$< 1e-4$	0.197
Error	2.626	952	0.003			
Total	47.098	1199				

From the results on TREC-COVID we observe that both the topic and its formulations have a large effect. The importance of the formulation factor indicates that, with an appropriate topic formulation, the performance on the topic can change greatly. Similar to what we observed in Table 2, the interaction between the topic and the models category is large ($\omega^2 = 0.390$), larger than the effect of both the sole category and the model. Also the interaction between the topic formulation and the models category is large ($\omega^2 = 0.197$), although not as large as the one between topic and category. This suggests that the semantic gap relates more to the underlying information need than the different topic formulations.

Overall, we hypothesize that the relation between topics and model categories, highlighted by ANOVA, links to the semantic gap and the association of a topic with its relevant documents. For instance, if a topic has many relevant documents containing synonyms of the query terms, then a semantic model might be best suited to perform retrieval. In fact, in this case, most of the topic formulations will not contain all the possible query synonyms and will thus be affected by the semantic gap. Conversely, topics that can be easily represented by few keywords – likely to be found within relevant documents – will have less ambiguous formulations, which are best suited to lexical models.

3.2 RQ2: Features Importance for the Semantic Gap

Section 3.1 showed the impact of choosing the proper models category depending on the query at hand. If we could classify queries as semantically hard or easy, we might also adopt an IR model from the right category. To properly train a classifier capable of doing that, we need *i)* to label queries as "semantic" or "lexical", and *ii)* to find a set of features that correlate with such aspects of the queries. The next two paragraphs tackle the above-mentioned challenges.

Labeling queries. The first aspect we address is the labeling of queries as "semantic" or "lexical". The absence of a rigorous definition of *semantically hard* or *easy* for a query prevents us from manually labeling queries as "semantic" or "lexical". Therefore, we propose to label queries according to how the two models categories perform on them. To the best of our knowledge, this is the first automatic approach to address this problem.

To this end, we first compute the average performance of each model. Then, for each query, we perform the following three steps.

Table 4: OHSUMED queries classification.

Label	Confidence			
	$\alpha > 0.95$	$\alpha > 0.90$	$\alpha \leq 0.90$	**Total**
Semantic	13	3	10	26
Lexical	13	6	18	37
Both	26	9	28	63

Firstly, we compute for each model the relative improvement over its average performance. Secondly, we determine whether the relative improvement is, on average, greater for lexical or semantic models. Finally, we label the considered query as "semantic" if the improvement over the average model performance is greater for semantic models than for lexical ones; vice versa, we label the query as "lexical".

Note that we do not consider absolute performances to label queries, since even a poorly performing lexical method like TF-IDF (cfr. Table 1) might prove effective when the query is semantically easy. Thus, we focus on relative improvements, which provide more robust signals to performance outliers.

Let $\mathcal{S}$ be the set of models and Q the set of queries. We call $AP_s(q)$ the AP observed for the model s on the query q, and $\text{MAP}_s(Q)$ and $\text{std}_s(Q)$ respectively the MAP and the standard deviation of the AP observed for the model s over the queries Q. We define $Z_{s,q} = \frac{AP_s(q) - \text{MAP}_s(Q)}{\text{std}_s(Q)}$ the relative improvement over the mean performance.

By standardizing relative improvements, we account for the variability in models performances. Then, let $\mathcal{S}_s$ be the set of semantic models, and $\mathcal{S}_l$ the set of lexical models.

Definition 3.1. A query q is labeled as "semantic" iff

$$\frac{\sum_{s \in \mathcal{S}_s} Z_{s,q}}{|\mathcal{S}_s|} >_\alpha \frac{\sum_{s \in \mathcal{S}_l} Z_{s,q}}{|\mathcal{S}_l|},$$

where $>_\alpha$, with $\alpha \in [0.5, 1)$, indicates that the mean relative improvement for semantic models is statistically significantly higher than that for lexical models at significance level α. Queries are labeled as "lexical" using the opposite ordering relation ($<_\alpha$).

Therefore, using the above-mentioned definition we can label queries as either "semantic" or "lexical" at a specific level of α. In practice, given a query q, we call $\mathcal{Z}_{q,sem} = \{Z_{s,q} \; \forall \; s \in \mathcal{S}_s\}$ the set of relative improvements of the semantic models for q, and $\mathcal{Z}_{q,lex} = \{Z_{s,q} \; \forall \; s \in \mathcal{S}_l\}$ the set of relative improvements of the lexical models for q. Using an unpaired t-test, we determine whether $\mathcal{Z}_{q,sem}$ has greater mean than $\mathcal{Z}_{q,lex}$. If so, then q is labeled as "semantic". On the other hand, if $\mathcal{Z}_{q,lex}$ has statistically significantly greater mean than $\mathcal{Z}_{q,sem}$, then q is labeled as "lexical". Otherwise, q is labeled as "neutral".

Tables 4 and 5 report the statistics of our labeling approach for OHSUMED and TREC-COVID collections, respectively, at different levels of confidence. We can observe that, in both collections, queries labeled with confidence above $\alpha = 0.90$ ($p\text{-}value < 0.1$) make up more than half of the total queries (i.e., 55.6% and 57.5% respectively). Another interesting observation is that queries labeled

Table 5: TREC-COVID queries classification.

Label	Confidence			
	$\alpha > 0.95$	$\alpha > 0.90$	$\alpha \leq 0.90$	**Total**
Semantic	27	7	26	60
Lexical	27	8	25	60
Both	54	15	51	120

with high confidence split evenly between lexical and semantic categories. This confirms what we observed in Tables 2 and 3, where the effect of the sole category plays a marginal role on performance. Focusing on TREC-COVID queries, we observe that different formulations of the same topic are either classified always in the same category or, when this is not the case, such formulations are labeled with low confidence[1]. This further explains the magnitude of the effects observed in Table 3, where the topic formulation showed a lower, although significant, interaction with the models category compared to that of the topic. The only exceptions are topics 16 and 23, where the *narrative* formulation is lexical while *concatenation* and *query*, for topic 16, and *concatenation*, *description*, and *query*, for topic 23, are semantic with confidence > 0.95. In this regard, it is interesting to note that, for both topics, the formulation labeled as "lexical" is always the *narrative* one. We attribute the reason for this to the richer linguistic structure of the *narrative* formulation, which, in both topics, presents a better description, as well as several relevant concepts, of the underlying information need – thus limiting the semantic gap and reducing the need for semantic models.

In the following, we restrict to queries labeled with confidence above 0.90, as we want to focus on queries that have been labeled with a high degree of confidence. Moreover, queries labeled as "neutral" for $\alpha = 0.90$ have been discarded.

Features and Category Selection. To address the second aspect of **RQ2** – that is, classifying a query as "semantic" or "lexical" – we explore two different sets of pre-retrieval features: Lexical- and Semantic-oriented features. Lexical-oriented features are based on query and corpus statistics and depend on the distribution of terms within the collection. Regarding semantic-oriented features, we first perform semantic indexing on OHSUMED and TREC-COVID collections as in [1]. Then, we adopt features similar to those proposed by Mothe and Tanguy [28], but, instead of considering only query-based features, we take into account both query- and corpus-based features. The considered features are reported and described in Table 6.

We employ three well-known classification models to understand the effectiveness of the considered pre-retrieval features when used to classify queries into lexical and semantic categories. The adopted models are: Decision Tree (`DTr`), Support Vector Machine (`SVM`), and Multi-Layer Perceptron (`MLP`). To perform experiments, we label queries using the process described above and we restrict to "semantic" and "lexical" queries that present a significance score greater than 0.90. For each classifier, we perform grid search with cross-validation to obtain the best hyper-parameters. We adopt 5-fold cross-validation for TREC-COVID, whereas we use 3-fold cross-validation for OHSUMED to avoid obtaining single-class folds due to the low number of samples. The results of the different classifiers are reported in Table 7, where we report mean and standard deviation over the different folds. To determine results significance (marked as †), we apply a randomization test with Bonferroni correction for multiple comparisons [32].

Regarding OHSUMED, we first highlight that `MLP` is the best performing method. However, `MLP` is also the method with the largest standard deviation for F1. This is likely due to the small number of samples – i.e., 35 queries labeled with confidence above 0.90. On top of this, none of the considered methods perform statistically better than the random classifier. Conversely, results for TREC-COVID are more stable – highlighting the impact the number of samples has on the stability of the classifiers performance. Also in TREC-COVID, both `SVM` and `MLP` are not statistically better than the random classifier. On the other hand, however, `DTr` obtains preliminary yet promising performance (i.e., 67% for accuracy and 66% for F1) and it is significantly better than the random classifier for both measures. This suggests the presence of underlying patterns within data and the potential of the considered features to distinguish between semantically hard ("semantic") and easy ("lexical") queries.

Relying on the results of the decision tree, we further investigate the features importance to determine which features correlate the most with the semantic gap, causing the query to be either semantically easy or hard. We only consider the decision tree built for TREC-COVID, since results on OHSUMED are not statistically significant. The first two features by importance are `QDF` (number of documents containing at least one query term) and `WSDF` (number of documents containing only query terms and no synonyms). Their importance is, respectively, 17.6% and 16.7%. These features are both related to the distribution of the query terms in the collection. For this reason, they are likely used by the classifier to identify semantically easy queries. Indeed, a large number of documents containing query terms is a potential indicator for the performance of lexical models. Besides, the fact that `WSDF` is the second most important feature is a further evidence of this: if several documents contain query terms, but only few of them present also synonyms of such terms, then the semantic gap will likely be small and lexical models will be effective. The third feature by importance is `meanSCQ` (12.1%): a pre-retrieval score based on IDF. A query having a high `meanSCQ` score indicates that lexical models are likely to perform well. This is due to the fact that most of the lexical approaches rely on heuristics based on IDF. Note also that `SCQ` is considered a "low performing" feature for predicting queries performance [11]. Nevertheless, in our scenario, it gains relevance in determining which models category performs best for the query. The fourth feature is `stdNCPQT` (the standard deviation over the number of concepts for each polysemous word in the query). This feature has importance 10.1%, which indicates the relevance of polysemy in determining the models category: having (several) query words with different concepts associated makes the query ambiguous and semantic models best suited to address it.The two subsequent features are `sumNSEQC` (8.8%) and `maxNSEQC` (7.3%). They represent,

[1] we omit these statistics, due to space reasons

Table 6: Pre-retrieval features considered for the category selection task.

Name	Description
	Lexical-oriented features
QL	Number of terms in the query [28]
{std,mean,max}IDF	Features based on the distribution of the IDF over the query terms [17]
{sum,mean,max}SCQ	Features based on the similarity between corpus [42]
QDF	Number of documents containing at least one query term
	Semantic-oriented features
QPD	Number of polysemous words within the query
{sum,std,max}NCQT	Sum, standard deviation, and max over the number of concepts related to query terms
{sum,std}NCPQT	Sum and standard deviation over the number of concepts related to polysemous query terms only
QSD	Number of synonymous words within the query
{sum,std,max}NSEQC	Sum, standard deviation, and max over the number of different synset elements related to query concepts
{sum,std}NSQC	Sum and standard deviation over the number of different synonyms related to query concepts
SDF	Number of documents containing at least one synonym of a query term
WSDF	Number of documents containing at least one query term and no synonyms of the query terms
WTDF	Number of documents containing at least one query synonym and no query terms

Table 7: Classifiers performance. We report mean and standard deviation over 3- and 5-folds for OHSUMED and TREC-COVID, respectively. † indicates statistical significance over the random classifier, according to a permutation test with significance 0.95 **and Bonferroni correction.**

	OHSUMED		TREC-COVID	
	Accuracy	**F1**	**Accuracy**	**F1**
`DTr`	0.626 (0.089)	0.586 (0.057)	**0.668 (0.093)**†	**0.659 (0.141)**†
`SVM`	0.687 (0.074)	0.611 (0.079)	0.623 (0.053)	0.610 (0.136)
`MLP`	**0.740 (0.081)**	**0.675 (0.146)**	0.628 (0.217)	0.590 (0.269)

respectively, the sum and the maximum of the number of synset elements related to the query concepts. Both features are related to synonymy, which is another relevant aspect that identifies the presence of the semantic gap between queries and documents. Similarly to our intuition about polysemy, having query words with several synonyms suggests that semantic models are best suited to retrieve relevant documents. Other features with decreasing, but significant, importance are `SDF` (5.7%) and `sumNCPQT` (5.2%). As for the remaining features, they are negligible according to the classifier.

Thus, even though the results are preliminary and indicate there is large room for improvement, they still highlight that the considered lexical- and semantic-oriented features relate with models categories. Therefore, they can be used as a starting point to investigate the presence of the semantic gap within test collections and to build better approaches for category selection.

4 CONCLUSION

In this work, we investigated the impact of the semantic gap on query performance, which features can be used to determine this gap, and whether we can exploit them to classify query as semantically easy ("lexical") or hard ("semantic"). Using ANOVA, we quantified the interaction between topics, queries, and models categories. The results showed that such interaction is large, highlighting the importance of choosing the proper models category for retrieval performance. Surprisingly, the analyses indicated that topics interact more than queries with models categories. This suggests that the semantic gap relates more to the underlying information need than the different topic formulations. Then, we proposed a labeling strategy, based on relative improvements, to annotate queries as "semantic" or "lexical". Finally, we explored two different sets of pre-retrieval features and we deployed a prototype classifier to understand the effectiveness of such features when used to classify queries. We obtained promising results, which suggest a correlation between the considered features and the models categories.

As future work, we plan to further explore features extraction and selection. In this regard, the preliminary results suggested that the considered features relate to models categories, but also highlighted that such correlation is weak and needs to be improved. Beyond pre-retrieval features, we also plan to investigate features related to retrieval models – thus getting closer to a post-retrieval setup. In this sense, we plan to adopt a pseudo-relevance strategy that considers retrieved documents and looks at the distribution of lexical- and semantic-oriented features in such documents. Finally, we plan to consider other domains besides the medical one, such as the news or Web domains.

ACKNOWLEDGMENTS

The work was partially supported by the ExaMode project, as part of the European Union H2020 program under Grant Agreement no. 825292.

REFERENCES

[1] M. Agosti, S. Marchesin, and G. Silvello. 2020. Learning Unsupervised Knowledge-Enhanced Representations to Reduce the Semantic Gap in Information Retrieval. *ACM Trans. Inf. Syst.* 38, 4 (2020), 38:1–38:48.

[2] G. Amati and C. J. van Rijsbergen. 2002. Probabilistic Models of Information Retrieval based on measuring the Divergence From Randomness. *ACM Trans. Inf. Syst* 20, 4 (2002), 357–389.

[3] J. A. Aslam and V. Pavlu. 2007. Query Hardness Estimation Using Jensen-Shannon Divergence Among Multiple Scoring Functions. In *Proc. of the 29th European Conference on IR Research, ECIR 2007, Rome, Italy, April 2-5, 2007.* 198–209.

[4] N. Balasubramanian. 2011. *Query-Dependent Selection of Retrieval Alternatives.* Ph.D. Dissertation. University of Massachusetts Amherst.

[5] N. Balasubramanian and J. Allan. 2010. Learning to Select Rankers. In *Proc. of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, July 19-23, 2010.* ACM, 855–856.

[6] D. Banks, P. Over, and N.-F. Zhang. 1999. Blind Men and Elephants: Six Approaches to TREC data. *Information Retrieval* 1, 1-2 (1999), 7–34.

[7] W. B. Croft, D. Metzler, and T. Strohman. 2009. *Search Engines: Information Retrieval in Practice.* Addison-Wesley, Reading (MA), USA.

[8] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. 2002. Predicting Query Performance. In *Proc. of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August 11-15, 2002, Tampere, Finland.* 299–306.

[9] T. Edinger, A. M. Cohen, S. Bedrick, K. H. Ambert, and W. R. Hersh. 2012. Barriers to Retrieving Patient Information from Electronic Health Record Data: Failure Analysis from the TREC Medical Records Track. In *AMIA 2012, American Medical Informatics Association Annual Symposium.* AMIA.

[10] G. Faggioli and N. Ferro. 2021. System Effect Estimation by Sharding: A Comparison Between ANOVA Approaches to Detect Significant Differences. In *Proc. of the 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 - April 1, 2021.* Springer International Publishing, Cham, 33–46.

[11] G. Faggioli, O. Zendel, J. S. Culpepper, N. Ferro, and F. Scholer. 2021. An Enhanced Evaluation Framework for Query Performance Prediction. In *Proc. of the 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 - April 1, 2021.* Springer International Publishing, Cham, 115–129.

[12] N. Ferro and D. Harman. 2009. CLEF 2009: Grid@CLEF Pilot Track Overview. In *Proc. CLEF.*

[13] N. Ferro and G. Silvello. 2018. Toward an Anatomy of IR System Component Performances. *J. Assoc. Inf. Sci. Technol.* 69, 2 (2018), 187–200.

[14] A. Griffiths, H. C. Luckhurst, and P. Willett. 1986. Using Interdocument Similarity Information in Document Retrieval Systems. *J. Am. Soc. Inf. Sci.* 37, 1 (1986), 3–11.

[15] C. Hauff, D. Hiemstra, and F. de Jong. 2008. A Survey of Pre-Retrieval Query Performance Predictors. In *Proc. CIKM.* 1419–1420.

[16] B. He and I. Ounis. 2004. A Query-based Pre-retrieval Model Selection Approach to Information Retrieval. In *Proc. of Computer-Assisted Information Retrieval (Recherche d'Information et ses Applications) - RIAO 2004, 7th International Conference, University Of Avignon, France, April 26-28, 2004.* CID, 706–719.

[17] B. He and I. Ounis. 2004. Inferring Query Performance Using Pre-retrieval Predictors. In *Proc. of the String Processing and Information Retrieval, 11th International Conference, SPIRE 2004, Padova, Italy, October 5-8, 2004.* 43–54.

[18] W. Hersh, C. Buckley, T. J. Leone, and D. Hickam. 1994. OHSUMED: An Interactive Retrieval Evaluation and New Large Test Collection for Research. In *Proc. of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval. Dublin, Ireland, 3-6 July 1994.* Springer London, London, 192–201.

[19] İ . Kocaba ş, B. T. Din ç er, and B. Karao ğ lan. 2014. A nonparametric term weighting method for information retrieval based on measuring the divergence from independence. *Information Retrieval* 17, 2 (2014), 153–176.

[20] B. Koopman and G. Zuccon. 2014. Why Assessing Relevance in Medical IR is Demanding. In *Proc. of the Medical Information Retrieval Workshop at SIGIR co-located with the 37th annual international ACM SIGIR conference (ACM SIGIR 2014) (CEUR Workshop Proceedings, Vol. 1276).* CEUR-WS.org, 16–19.

[21] B. Koopman, G. Zuccon, P. Bruza, L. Sitbon, and M. Lawley. 2016. Information retrieval as semantic inference: a Graph Inference model applied to medical search. *Inf. Retr. Journal* 19, 1-2 (2016), 6–37.

[22] O. Levi, F. Raiber, O. Kurland, and I. Guy. 2016. Selective Cluster-Based Document Retrieval. In *Proc. of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016.* ACM, 1473–1482.

[23] H. Li and J. Xu. 2014. Semantic Matching in Search. *Found. Trends Inf. Retr.* 7, 5 (2014), 343–469.

[24] X. Liu and W. B. Croft. 2006. *Experiments on retrieval of optimal clusters.* Technical Report. University of Massachusetts Amherst.

[25] X. Liu, J. Y. Nie, and A. Sordoni. 2016. Constraining Word Embeddings by Prior Knowledge - Application to Medical Information Retrieval. In *Proc. of the 12th Asia Information Retrieval Societies Conference, AIRS 2016.* Springer, 155–167.

[26] S. Marchesin, A. Purpura, and G. Silvello. 2020. Focal elements of neural information retrieval models. An outlook through a reproducibility study. *Inf. Process. Manag.* 57, 6 (2020), 102109.

[27] T. Mikolov, K. Chen, G. Corrado, and J. Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proc. of the 1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013.*

[28] J. Mothe and L. Tanguy. 2005. Linguistic Features to Predict Query Difficulty. In *Proc. of the Predicting query difficulty-methods and applications workshop, co-located with the ACM Conference on research and Development in Information Retrieval, SIGIR 2005.* 7–10.

[29] S. E. Robertson and U. Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trnd. Inf. Retr.* 3, 4 (2009), 333–389.

[30] H. Roitman. 2018. Query Performance Prediction using Passage Information. In *Proc. of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018.* 893–896.

[31] A. Rutherford. 2011. *ANOVA and ANCOVA. A GLM Approach* (2nd ed.). John Wiley & Sons, New York, USA.

[32] P. Sedgwick. 2012. Multiple significance tests: the Bonferroni correction. *Bmj* 344 (2012).

[33] A. Shtok, O. Kurland, and D. Carmel. 2016. Query Performance Prediction Using Reference Lists. *ACM Trans. Inf. Syst* 34, 4 (2016), 19:1–19:34.

[34] A. Tombros, R. Villa, and C. J. van Rijsbergen. 2002. The Effectiveness of Query-Specific Hierarchic Clustering in Information Retrieval. *Inf. Process. Manag.* 38, 4 (2002), 559–582.

[35] C. Van Gysel, M. De Rijke, and E. Kanoulas. 2018. Neural vector spaces for unsupervised information retrieval. *ACM Trans. Inf. Syst.* 36, 4 (2018), 1–25.

[36] E. Voorhees, T. Alam, S. Bedrick, D. Demner-Fushman, W. R. Hersh, K. Lo, K. Roberts, I. Soboroff, and L. L. Wang. 2021. TREC-COVID: Constructing a Pandemic Information Retrieval Test Collection. *SIGIR Forum* 54, 1, Article 1 (Feb. 2021), 12 pages.

[37] E. Voorhees, D. Samarov, and I. Soboroff. 2017. Using Replicates in Information Retrieval Evaluation. *ACM Trans. Inf. Syst* 36, 2 (2017), 12:1–12:21.

[38] I. Vulić and M. F. Moens. 2015. Monolingual and Cross-Lingual Information Retrieval Models Based on (Bilingual) Word Embeddings. In *Proc. of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval.* ACM, 363–372.

[39] L. L. Wang, K. Lo, Y. Chandrasekhar, R. Reas, J. Yang, D. Burdick, D. Eide, K. Funk, Y. Katsis, R. Kinney, Y. Li, Z. Liu, W. Merrill, P. Mooney, D. Murdick, D. Rishi, J. Sheehan, Z. Shen, B. Stilson, A. Wade, K. Wang, N. X. Ru Wang, C. Wilhelm, B. Xie, D. Raymond, D. S. Weld, O. Etzioni, and Sebastian Kohlmeier. 2020. CORD-19: The COVID-19 Open Research Dataset. arXiv:2004.10706 [cs.DL]

[40] C. Zhai. 2008. Statistical Language Models for Information Retrieval. A Critical Review. *Found. Trnd. Inf. Retr.* 2, 3 (2008), 137–213.

[41] R. Zhao and W. I. Grosky. 2002. Narrowing the semantic gap - improved text-based web document retrieval using visual features. *IEEE Trans. Multimedia* 4, 2 (2002), 189–200.

[42] Y. Zhao, F. Scholer, and Y. Tsegay. 2008. Effective Pre-retrieval Query Performance Prediction Using Similarity and Variability Evidence. In *Proc. of the 30th European Conference on IR Research, ECIR 2008, Glasgow, UK, March 30-April 3, 2008.* 52–64.