# Assigning Belief Scores to Names in Queries

Christopher Dozier

Research and Development
Thomson Legal and Regulatory
610 Opperman Drive
Eagan, MN 55123,USA
chris.dozier@westgroup.com

## ABSTRACT

Assuming that the goal of a person name query is to find references to a particular person, we argue that one can derive better relevance scores using probabilities derived from a language model of personal names than one can using corpus based occurrence frequencies such as inverse document frequency (idf). We present here a method of calculating person name match probability using a language model derived from a directory of legal professionals. We compare how well name match probability and idf predict search precision of word proximity queries derived from names of legal professionals and major league baseball players. Our results show that name match probability is a better predictor of relevance than idf. We also indicate how rare names with high match probability can be used as virtual tags within a corpus to identify effective collocation features for person names within a professional class.

## 1. INTRODUCTION

Some of the most common types of queries submitted to search engines both on the internet and on proprietary text search systems consist simply of a person's name. To improve the way such queries are handled, it would be useful if search engines could estimate the likelihood or belief that a name contained in a document pertains to the name in the query. Traditionally, relevance likelihood for name phrases has been based on inverse document frequency or idf, [3][4]. The idea behind this relevance estimate is that names which rarely occur in the corpus are thought to be more indicative of relevance than names that commonly occur.

Assuming that the goal of a person name query is to find references to a particular person, we argue that one can derive better relevance scores using probabilities derived from a language model of personal names than one can using corpus based occurrence frequencies. The reason for this is that finding references to a particular person in text is more dependent upon the relative rarity of the name with respect to the human population than it is on the rarity of the name within a corpus.

To get an intuitive idea of this point, consider that, within a corpus of 27,000 Wall Street Journal articles published between January and August of the year 2000, the name "Trent Lott" occurred in 80 documents while the name "John Smith" occurred in 24. All 80 references to "Trent Lott" referred to the majority leader of the U.S. Senate, while "John Smith" references mapped to 5 different people. This is not surprising. From our experience, we know that "Trent Lott" is an uncommon name and "John Smith" is a common one.

We present here evidence that name match probability based on a language model predicts relevance for name queries far better than idf. It may be argued that idf was never intended to be used to measure the relative ambiguity of a name query. However, idf is the standard measure used in probabilistic search engines to measure the degree of relevance terms and phrases within a collection have to the terms and phrases in queries, [1] [5]. For this reason, we take idf to be the standard against which to compare name match probability.

Being able to predict relevance through name match probabilities enables us to do three things. First, it tells us when we need to add information to the query to improve precision either by prompting the user for it or automatically expanding the query. Second, and perhaps more importantly, it enables us to use names with high match probabilities as virtual tags that can help us find useful collocation features to disambiguate names within a given class of names, such as the names of attorneys and judges. For purposes of this paper, we define an ambiguous name as one likely to be shared by many people and an unambiguous name as one likely to apply to a single person or to only a few people. And third, match probability can be used as a feature within a name search operator to improve search precision.

## 2. DESCRIPTION OF MATCH PROBABILITY CALCULATION FOR PERSON NAMES

The motivation for our work is an effort to develop a name search operator to find attorneys and judges in the news. In our particular application, we wish to allow users to search for newspaper references to attorneys and judges listed in a directory of U.S. legal professionals. This directory contains the curriculum vitae of approximately one million people. In this section, we show how we calculate person name match probability.

To compute the probability of relevance or match probability for a name, we perform three steps. First, we compute a probability distribution for the first and last names in our name directory. This is our language model. Second, we compute a name's probability by multiplying its first name probability with its last name probability. Third, we compute its match probability by taking the reciprocal of the product of the name probability and the size of the human population likely to be referenced in the corpus. For our Wall Street Journal test corpus, we estimated this size to be approximately the size of the U.S. population or 300 million. Formulas for the three steps are shown below.

$$(1)\quad P(first\_name) = \frac{F}{N}$$

$$P(last\_name) = \frac{L}{N}$$

where F = number of occurrences of first name, L = number of occurrences of last name, and N = number of names in the directory.

$$(2)\quad P(name) = P(first\_name) \cdot P(last\_name)$$

$$(3)\quad P(name\_match) = \frac{1}{\left(H \cdot P(name)\right) + 1}$$

where H = size of human population likely to be referenced by the collection.

Example calculations for Trent Lott and John Smith are shown below in Table 1.

In this example, the match probability for Trent Lott is approximately four orders of magnitude higher than the match probability for John Smith, while idf or document frequency suggests the likelihood of relevance for documents retrieved for John Smith is higher than for documents retrieved for Trent Lott. Both empirically and intuitively, match probability is a better predictor of relevance here than idf.

## 3. EVALUATION OF NAME MATCH PROBABILITY VERSUS IDF

To test our hypothesis that name match probability predicts relevance better than idf, we compared how well name queries with high match probabilities performed against name queries with high idf. We performed two experiments. In the first, we selected names of individuals in our legal directory. In the second, we used the names of currently active major league baseball players.

To conduct the first experiment, we labeled person names in a collection of 27,000 WSJ documents with a commercially available name tagging program. We then extracted these names and created a merged list of names specified by first and last name and pulled from this list names that occurred within our legal directory. We then sorted this list by name match probability and by document occurrence frequency (which is equivalent to idf) to create two lists. We then binned the names in the name match probability list into sets that fell between the following probability ranges: 1.0-0.9, 0.9-0.8 ,0.8-0.7, 0.7-0.6, 0.6-0.5, 0.5-0.4, 0.4-0.3, 0.3-0.2, 0.2-0.1, and 0.1-0.0. We binned the names in the document frequency list into sets that fell into the following document occurrence frequencies: 1, 2, 3, 4, 5, 6, 7, 8, 9, and >=10.

We then selected 50 names at random from each of these bins (except for bins associated with 0.8-0.7 and 0.7-0.6 probabilities which contained 42 and 31 names respectively). For each name selected, we identified the legal directory entry that was compatible with the name. In most cases, only one legal directory entry was compatible with the name. In some cases, multiple entries were compatible. For example, the name “Paul Brown” is compatible with 71 legal directory entries since there are 71 people in the directory with the first name “Paul” and the last name “Brown”. In these cases, we selected one of the entries at random.

For each name in each bin, we found the set of documents in the WSJ collection that would be returned by the word proximity query “First_name +2 Last_name”. That is, the documents that contained the first name followed within two words by the last name.

The search precision results for match probability and document frequency bins are shown in tables 2 and 3 below. The search precision of each bin was the number of relevant documents returned by the names in the bin divided by the total number of documents returned. The row labeled “Number Unique Names in Each Category” is a count of the number of unique first and last name pairs found within the WSJ collection for the probability and document frequency ranges indicated. It was from these sets of names that we selected our queries.

The results in tables 2 and 3 show that match probability does a better job of estimating relevance than idf. Table 2 shows that search precision goes up as match probability rises. Table 3 shows no apparent correspondence between document frequency and search precision.

**Table 1: Example Calculation of Match Probability**

| Name | P(first name) | P(last name) | P(name) | P(name match) | Doc Freq |
|---|---|---|---|---|---|
| Trent Lott | 0.000084 | 0.000048 | 0.00000000408 | 0.449371705 | 80 |
| John Smith | 0.036409 | 0.006552 | 0.00023857 | 0.00001397 | 24 |

**Table 2: Search Precision At Different Match Probabilities for Names Compatible with Judge and Attorney Names for WSJ Collection**

| Match Prob Range | 1.0 - 0.9 | 0.9 – 0.8 | 0.8 – 0.7 | 0.7 – 0.6 | 0.6 – 0.5 | 0.5 – 0.4 | 0.4 – 0.3 | 0.3 – 0.2 | 0.2 – 0.1 | 0.1 – 0.0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Search Precision | 0.835 | 0.754 | 0.595 | 0.677 | 0.596 | 0.708 | 0.628 | 0.544 | 0.520 | 0.12 |
| Number Unique Names in Each Category | 80 | 61 | 42 | 31 | 57 | 72 | 113 | 135 | 292 | 10758 |

**Table 3: Search Precision At Different Document Occurrence Frequencies for Names Compatible with Judge and Attorney Names for WSJ Collection**

| Doc Freq | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | >=10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Search Precision | 0.18 | 0.10 | 0.10 | 0.20 | 0.06 | 0.10 | 0.08 | 0.18 | 0.14 | 0.24 |
| Number Unique Names in Each Category | 7702 | 1946 | 703 | 374 | 224 | 145 | 95 | 75 | 55 | 322 |

**Table 4: Search Precision At Different Match Probabilities for Names Compatible with Names of Major League Baseball Players for WSJ Collection**

| Match Prob Range | 1.0 - 0.9 | 0.9 – 0.8 | 0.8 – 0.7 | 0.7 – 0.6 | 0.6 – 0.5 | 0.5 – 0.4 | 0.4 – 0.3 | 0.3 – 0.2 | 0.2 – 0.1 | 0.1 – 0.0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Search Precision | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 1.0 | 0.939 | 0.633 |
| Number Unique Names in Each Category | 15 | 5 | 2 | 2 | 2 | 3 | 2 | 7 | 7 | 48 |

**Table 5: Search Precision At Different Document Occurrence Frequencies for Names Compatible with Names of Major League Baseball Players for WSJ Collection**

| Doc Freq | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | >=10 |
|---|---|---|---|---|---|---|---|---|---|---|
| Search Precision | 0.888 | 0.882 | 0.952 | 1.0 | 0.75 | 0.666 | 1.0 | NA | 1.0 | 0.74 |
| Number Unique Names in Each Category | 45 | 17 | 7 | 3 | 4 | 6 | 2 | 0 | 1 | 8 |

In the second experiment, we performed basically the same steps described above on the names of the 286 baseball players currently playing in the major leagues. We assigned name match probabilities to these names using the language model we derived from the legal directory. Of the 286 names, we found 82 that were compatible with one or more name instances in the WSJ collection. For all 82, we found the set of documents in the WSJ collection that would be returned by the word proximity query "First_name +2 Last_name". We then measured how frequently the documents returned for a particular word proximity query actually referenced the player with which the name query was paired. As in the attorney and judge name experiment, name match probability predicted relevance more accurately than idf. The results for baseball player names are shown in tables 4 and 5 above.

Note that on average the search precision for baseball players was higher than for attorneys and judges. This is due to the combined effects of there being far fewer baseball player names than attorney and judge names and the fact that the average probability of a baseball player being mentioned in the news is higher than the average probability for a judge or attorney being mentioned.

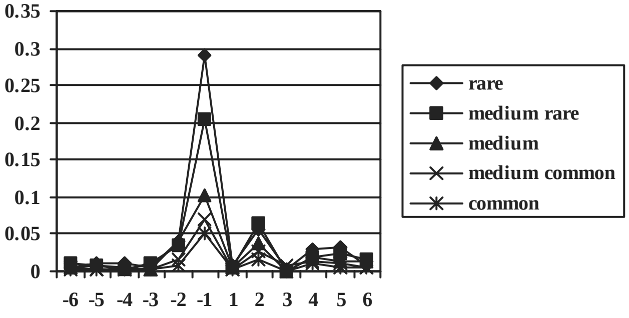

**Figure1: Conditional probability of attorney terms by word position relative to name**

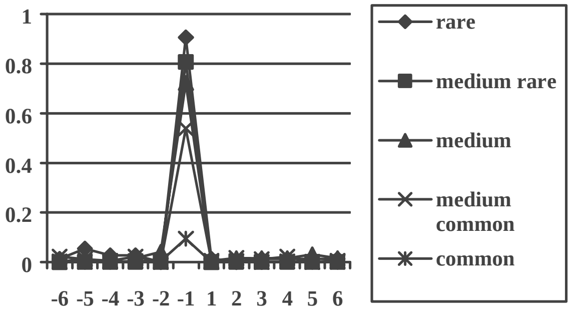

**Figure2: Conditional probability of judge terms by word position relative to name**

## 4. USING RARE NAMES TO IDENTIFY SEARCH FEATURES

An important use of name match probabilities is the identification of co-occurrence features in text that can serve to disambiguate name references. If we know certain names in the corpora very probably refer to certain individuals listed in a professional directory, we can look for words that co-occur frequently with these names but infrequently with names in general. These words are likely to work well at disambiguating references to names of low match probability.

As an example of feature identification, consider the figures 1 and 2 above. In these figures, the word "rare" stands for the 20% of names in the legal directory that have the highest match probability. The phrase "medium rare" stands for the next 20% and so on. The word "common" then stands for the 20% of names with the lowest match probability. For each of the five categories of name rarity, the graphs in the figures show the probability of an appositive term occurring at a given word position relative to the position of a name.

Figure 1 shows the probability of attorney appositive nouns such as "attorney", "lawyer", "counsel", or "partner" occurring at 12 different word positions around attorney names of varying degrees of rarity. Position –1 stands for the word position directly before the name. Position +1 stands for the position directly after. Position –2 stands for the word position two words in front of the name and so on. Figure 2 shows the probability of judge appositive nouns such as "judge" or "justice" occurring around judge names.

The graphs in figures 1 and 2 show that the probability of appositive terms occurring at particular word positions grows steadily as the name rarity increases. This demonstrates that appositive terms are good indicators for judge and attorney names within the WSJ collection. The figures also shows the word positions in which we should look for appositive terms.

Figure 1 shows that we should look for attorney appositives in word positions –2, -1, +2, +4, and +5. This makes intuitive sense because it accounts for sentence constructs such as those shown in table 6.

**Table 6: Examples of Use of Attorney Term Near Attorney Name**

| Relative Word Position | Example sentence |
|---|---|
| -2 | **Attorney** General Janet Reno said today ….. |
| -1 | **Attorney** Jack Smith defended his client vigorously. |
| +2 | said Vicki Patton, senior **attorney** for Environmental Defense |
| +4 | said Jim Hahn, Los Angeles City **Attorney** |
| +5 | says Buck Chapoton, a prominent Washington tax **attorney** |

The sudden drop off in appositive term probability at word position +1 also makes sense since an article, adjective, or other part of speech often occurs between a trailing appositive head noun and the proper noun it modifies. The drop off at word position +3 is still something of a mystery and is not something we can explain at this time. Since +3 behavior seems to have no linguistic basis that we can perceive, we do not rely on it in constructing our search operator.

Figure 2 shows that we should look for judge appositives in word position -1. This makes perfect sense since it accounts for constructs such as " Judge William Rehnquist" and "Justice Antonin Scalia". Figure 2 also suggests that using the -1 appositive test should yield good search recall since the conditional probability for rare names is about 0.9.

## 5. PRELIMINARY SEARCH OPERATOR EXPERIMENTS

We are currently investigating what levels of search precision and recall we can achieve with special attorney and judge name search operators using name rarity together with co-occurrence features such as appositive, city, state, firm, and court terms. Our preliminary results are encouraging. Initial experiments with the attorney search operator indicate we can achieve a nine fold improvement in search precision over simple word proximity searches over the WSJ collection while sacrificing 18% recall. Preliminary results are shown in table 7 below. We produced these results by selecting 677 attorney names at random from the legal directory that existed within the WSJ collection. For each name, we ran word proximity searches using the first and last name of the lawyers and scored the results. Using the scored results from 377 of the names, we then trained a special Bayesian based name operator that used first name, last name, city, state, firm, and name rarity information as sources of name match evidence. Finally we tested the word proximity operator performance against the special name operator using the remaining 300 names.

Note that we have assumed above that word proximity searches yield 100% recall. This is not wholly accurate since it does not account for nicknames, use of first name initials, and so on. We plan to revise this recall estimate in the future, but for now we assume that a word proximity search on first and last name provides close to 100% recall in a collection such as the WSJ.

**Table 7: Comparison of Performance of Word Proximity Search and Special Name Operator Searches for Attorney Names**

| Search Method | Precision | Recall | F-measure |
|---|---|---|---|
| Word proximity | 0.09 | 1.00 | 0.17 |
| Attorney Name Search Operator | 0.85 | 0.82 | 0.83 |

## 6. FUTURE WORK

We plan to complete development of search operators for attorney and judges that make use of the combined features of name rarity, appositives, city, state, firm, and court terms. We plan to compare the performance of these operators against searches based on name indexes derived from combining MUC style extraction techniques and record linking techniques. [2] Our hope is that the search operators will perform at levels close to the indexed based searches so that we can avoid the operational costs of creating special name indexes.

We plan to mine names from text using name rarity and seed appositive phrases. For example, using a seed appositive phrase for a profession such as “expert witness”, we plan to identify and extract a set of expert witness names. From this initial set of names, we will identify rare names and use these to identify more appositive phrases. Once the appositive phrases are identified, we plan to extract more names, then more appositive phrases, and so on until a stopping condition is reached. In this manner, we hope to develop a technique to automatically extract name lists from text collections.

Finally we plan to assess whether it is possible to develop similar name match probability calculations for other types of names such as company names, organization names, and product names.

## 7. CONCLUSION

Assuming that the goal of a person name query is to find references to a particular person, we have shown that one can derive better relevance scores using probabilities derived from a language model of personal names than one can using corpus based occurrence frequencies. We presented here a method of calculating person name match probability using a language model derived from a directory of legal professionals. We compared how well name match probability and idf predict search precision of word proximity queries derived from names of legal professionals and major league baseball players. Our results showed that name match probability is a better predictor of relevance than idf. We also indicated how rare names with high match probability can be used as virtual tags within a corpus to identify effective collocation features for person names within a professional class.

## 8. REFERENCES

[1] Baeza-Yates, R. and Ribeiro-Neto, B., Modern Information Retrieval. ACM Press, New York, 1999.

[2] Dozier, C. and Haschart, R., "Automatic Extraction and Linking of Person Names in Legal Text" in Proceedings of RIAO '2000; Content Based Multimedia Information Access. Paris, France. pp.1305-1321. 2000

[3] de Lima, F. and Pedersen, J., Phrase Recognition and Expansion for Short, Precision-biased Queries based on a Query Log. In Proc.of the 22nd Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 145 – 152, Berkeley, California, USA, 1999.

[4] Thompson, P. and Dozier, C., Name Searching and Information Retrieval. In Proc.of the 2nd Conference on Empirical Methods in NLP, pp. 134 –140, Providence, Rhode Island, 1997.

[5] Turtle, H. and Croft, W., Inference Networks for Document Retrieval. In Proc.of the 13th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1 – 24, Brussels, Belgium, 1990.