The problem of this method is the size of confusion set C may be huge for multi-term queries. In practice, one term may have hundreds of possible candidates, then a query containing several terms may have millions. This might lead to impractical search and training using the maximum entropy modeling method. Our solution to this problem is to use *candidate pruning*. We first roughly rank the candidates based on the statistical n-gram language model estimated from query logs. Then we only choose a subset of C that contains a specified number of top-ranked (most probable) candidates to present to the maximum entropy model for offline training and online re-ranking, and the number of candidates is used as a parameter to balance top-line performance and run-time efficiency. This subset can be efficiently generated as shown in (Li et al., 2006).

5 Web Search Results based Query Spelling Correction

In this section we will describe in detail the methods for use of web search results in the query spelling correction task. In our work we studied two schemes. The first one only employs indicators of the input query's search results, while the other also looks at the most probable correction candidates' search results. For each scheme, we extract additional scheme-specific features from the available search results, combine them with baseline features and construct a new maximal model to perform candidate ranking.

5.1 Baseline model

We denote the maximum entropy model based on baseline model feature set as M0 and the feature set S0 derived from the latest state of the art works of (Li et al., 2006), where S0 includes the features mostly concerning the statistics of the query terms and the similarities between query terms and their correction candidates.

5.2 Scheme 1: Using search results for input query only

In this scheme we build more features for each correction candidate (including input query q itself) by distilling more evidence from the search results of the query. S1 denotes the augmented feature set, and M1 denotes the maximum entropy model based on S1. The features are listed as follows:

1. **Number of pages returned**: the number of web search pages retrieved by a web search engine, which is used to estimate the popularity of query. This feature is only for q.
2. **URL string**: Binary features indicating whether the combination of terms of each candidate is in the URLs of top retrieved documents. This feature is for all candidates.
3. **Frequency of correction candidate term**: the number of occurrences of modified terms in the correction candidate found in the title and snippet of top retrieved documents based on the observation that correction terms possibly co-occur with their misspelled ones. This feature is invalid for q.
4. **Frequency of query term**: the number of occurrences of each term of q found in the title or snippet of the top retrieved documents, based on the observation that the correct terms always appear frequently in their search results.
5. **Abbreviation pattern**: Binary features indicating whether inputted query terms might be abbreviations according to text patterns in search results.

5.3 Scheme 2: Using both search results of input query and top-ranked candidate

In this scheme we extend the use of search results both for query q and for top-ranked candidate c other than q determined by M1. First we submit a query to a search engine for the initial retrieval to obtain one set of search results R_q, then use M1 to find the best correction candidate c other than q. Next we perform a second retrieval with c to obtain another set of search results R_c. Finally additional features are generated for each candidate based on R_c, then a new maximum entropy model M2 is built to re-rank the candidates for a second time. The entire process can be schematically shown in Figure 3.

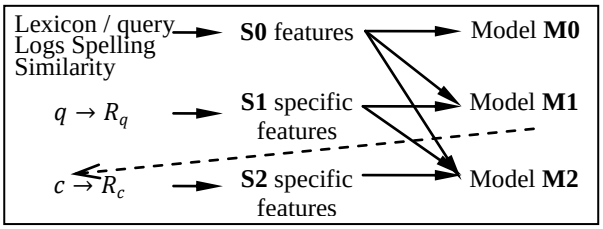

Figure 3. Relations of models and features