Table 1: Data Set Statistics

Name	Size	Query Stats				
		Mean	Min	Max	Median	Mode
USHMM Spoken Names	249	9.80	5	28	9	8
USHMM Names	656	63.77	7	164	54	60
Medical	9,883	10.58	2	30	10	10
US Census Sir Names	88,799	7.83	3	14	8	7

1 Evaluation

We evaluated Segments over 4 different data sets using various experimental methods. A detailed description follows.

1.1 Data Sets

We consider four data sets in our evaluation.

1. USHMM Names
2. USHMM Spoken Names
3. Medical Words
4. US Census Sir Names

Use of these data sets allow us to evaluate Segments over a variety of potential user inputs and environments. For example, the US Census Sir Names data set, which contains roughly 90,000 names, is a good evaluation of the scalability of Segments. Inversely, the USHMM Names data set is an evaluation over a small data set with many languages. Statistical descriptions of each data set can be found in Table 1. A detailed description of each data set along with justification for its use follows.

1.1.1 USHMM Spoken Names

This data set tests the performance of Segments using user query logs. Unlike other data sets which are synthetically altered (see 1.2.1) these contain real errors by users. We are unable to obtain “real world” user query logs, so we generated our own. See 1.2.2 for details on how we generated this data set. Due to the high cost of generating such a data set, this is the only data set based on user query logs.

1.1.2 USHMM Names

This data set is known as the Yizkor Books data set [2]. It is a collection of **14?** different languages. This data set demonstrates Segments ability to handle the structure of different European and Slovakian languages, on a small-scale.

1.1.3 Medical

This data set is take from [1]. It is a large collection of medical terms. The justification for this data set was to see how Segments performs in the medical domain, and with a larger data set (roughly 10,000 terms).

1.1.4 US Census Sir Names

This is a collection of Sir Names, sorted in order of frequency, according to the results of the U.S. Census bureau during their 1990 collection. The data set is available for public use [3]. The justification of this data set is to test the scalability of Segments.

1.2 Query Generation Methods

The data sets described in 1.1 fall into two categories: 1) Contain a list of correctly spelled queries or 2) Contain a list of incorrectly spelled queries which map to their correctly spelled counterpart. All data sets but USHMM Spoken Names are of the former form, and USHMM Spoken Names is the latter form. Those falling into the 1st category use synthetic method described in 1.2.1 to generate candidate search queries. Those falling into the 2nd category already have their candidate search queries (the user's misspelling of the target query work).

1.2.1 Synthetic

A candidate search query is defined as follows: for some query $q \in Q$, where Q is one of the data sets, $f(q) = q'$ where $f(q)$ is some synthetic function which alters q to generate q', called the candidate search query.

There are 4 different synthetic functions $f(q)$, that also take as input a magnitude of alteration $m \in M = \{1, 2, 3, 4\}$: $f(q, m)_{z \in F}$ where $F = \{$add_character, drop_character, replace_character, swap_character$\}$. This alteration is done for each tuple (q, f_z, m) where $f_z \in F$, $m \in M$, and $q \in Q$ to generate candidate search queries q' for every possible combination.

Consider the following example. Let $q =$ 'cat', $m = 2$, and $z =$'add_character'. Then $f(q, m)_z =$'c**j**at**s**'$= q'$ is *one possible* generation for q'. Note we emphasize possible since character and insertion (or replace/deletion/swap) index are randomly selected. A successful search is one where search algorithm s takes in q' and has the following equation hold: $(s(q') \cap \{q\}) = \{q\}$. If the equation holds true, the engine has found the correct result, otherwise not. The rank of q within the results of s is defined as i where $s(q') = \{x_1, x_2, ..x_n \mid x_i = q\}$. The results set $\{x_1..x_n\}$ is a descending sorted list where x_1 is the most likely to be our q according to s. Note it is possible that s does not find q, or $(s(q') \cap \{q\}) = \emptyset$, which means we do not check for the rank of q and it is not factored into the average rank for algorithm s.

1.2.2 Query Logs

To contrast the mechanical alteration of queries, we obtained user misspellings by the following process. First, we randomly selected a subset of 50 words from the USHMM Names data set. Next, we setup a test environment where a word is spoken to a user using Apple's text-to-speech command to speak the words to the user, who then types in their best guess of how to spell said word. This is done for each word in the data set (50 times). We had 5 users submit to this test, which resulted in 249 usable queries, with one query being removed because it was left blank. We justify the use of Apple's text-to-speech in place of a native speaker because our goal is to see how an average user would say a word, then have their friend type it back. For example, if you are transcribing words which are being dictated. We can further justify the fact that these words will likely be mispronounced because of the diversity of languages within the data set, and the inability to find someone capable of speaking each language.

1.3 Experimental Evaluation

2 Results

References

[1] http://www.spellingzone.com/medicalspelling.html

[2] M. Amir, *From Memorials to Invaluable Historical Documentation: Using Yizkor Books as Resource for Studying A Vanished World*, Annual Convention of the Association of Jewish Libraries, La Jolla, California, June 2001.

[3] United States Census Bureau 1990 Surnames, September 1, 2009. http://www.census.gov/genealogy/names/dist.all.last.