As the first two cases are trivial, we focus on case 3. In Inequalities 9a and 10, we will not miss records by bounding a record length when we consider $\mathbf{Q}[1]$. In other words, all the similar records that contain token $\mathbf{Q}[1]$ will be included in $\mathbb{C}$ or O, thus the potentially similar records outside $\mathbb{C} \cup O$ do not include $\mathbf{Q}[1]$, i.e., for $j = 1$ we have

$$W_2(R, Q) \geq \tau \wedge R \notin \mathbb{C} \cup O \Rightarrow \mathbf{Q}[j] \notin R \tag{28a}$$

$$\Leftrightarrow f(\mathbf{Q}[j], R) = 0. \tag{28b}$$

Similar to Inequalities 27a$\sim$27c in Appendix A.6, we have

$$W_2(R, Q) \geq \tau \Rightarrow \tau \mathsf{len}(R)\mathsf{len}(Q) \tag{29a}$$

$$\leq \sum_{t \preceq \mathbf{Q}[j]} w^2(t) f(t, R) f(t, Q) \tag{29b}$$

$$+ \sum_{t \succ \mathbf{Q}[j]} w^2(t) f(t, \max_{S \in \mathbb{B}} f(t, S)) f(t, Q). \tag{29c}$$

From Equation 28b we know Equation 29b is zero. Then we have $\mathsf{len}(R) \leq U_{\tau,Q,1} = U_r$ in case 3.

We continue this process by increasing j. In case 3 of each step, all the previous steps do not miss similar records, then all the similar records that contain at least one token in $\mathbf{Q}[: j]$ will be included in $\mathbb{C}$ or O. Any upcoming record that is similar to Q must not include any token in $\mathbf{Q}[: j]$ unless it has been included by $\mathbb{C}$ or O. Thus, Equation 28b always holds, and the expression given in Equation 29b is zero. We can claim $\mathsf{len}(R) \leq U_r$ is always true.

By maximizing the bounds of cases 1 and 3 in each step, we obtain Equation 17b. Similarly, we can prove Equation 17a. Thus Theorem 9 is proved.

B USABILITY OF SIMILARITY FUNCTIONS

This paper focuses on p-norm similarity given in Definition 3 with $p = 2$. To measure the relative similarity of two records, we formalize the correlation of two records by the summation of factor products based on their degrees and token weights, and generalize the tf-idf cosine similarity [10] with variant token weights and degrees.

B.1 Accuracy Comparisons

As for the precision comparison given in Section 2, we collected the data set from https://github.com/taolei87/askubuntu, which contained a preprocessed collection of questions taken from the AskUbuntu.com 2014 corpus dump. All its records had been tokenized and divided to two sets, in which each record had two fields "Title" and "Question body." Its training set had 167,765 records. In the testing set, each record had been annotated with several similar training records by domain experts. Based on the ground truth about similar pairs of these records, we did similarity search using different functions, and compared them using F1 score, as follows.

$$F1 = \frac{2 \times precision \times recall}{precision + recall}. \tag{30}$$

The following table shows the F1 cores of different functions based on "Title" and the union of both fields ("Full"). It is clear that the maximal F1 score of $2N_{\text{tfidf}}$ was comparable to $2N_{\text{idf}}$ (with a constant degree) based on the short "Title" field. When we used longer records ("Full"), the maximal F1 score of $2N_{\text{tfidf}}$ was better than that of $2N_{\text{idf}}$ by 5%. Notice that both of the weighted functions had their maximal F1 scores higher than Jaccard and Cosine.

Table 5: F1 of various functions with different thresholds.

	Title				Full			
τ	Jac	Cos	$2N_{\text{idf}}$	$2N_{\text{tfidf}}$	Jac	Cos	$2N_{\text{idf}}$	$2N_{\text{tfidf}}$
0.1	0.00	0.00	0.01	0.00	0.00	0.00	0.05	0.01
0.2	0.04	0.01	0.03	0.03	0.14	0.00	0.37	0.12
0.3	0.20	0.03	0.12	0.11	**0.40**	0.07	**0.42**	0.37
0.4	0.34	0.09	0.28	0.27	0.38	**0.39**	0.40	**0.47**
0.5	**0.38**	0.24	0.40	0.39	0.37	0.39	0.39	0.42
0.6	0.37	0.36	**0.42**	**0.42**	0.37	0.37	0.37	0.39
0.7	0.37	**0.38**	0.40	0.40	0.37	0.37	0.37	0.37
0.8	0.37	0.37	0.38	0.38	0.37	0.37	0.37	0.37
0.9	0.37	0.37	0.37	0.37	0.37	0.37	0.37	0.37

B.2 Discussions about Application Domains

As shown above, the token weights and degrees integrated in tf-idf 2-norm similarity show certain superiority in longer records. It becomes the Cosine function when we use the weight $w(t) = 1$ and use a boolean function for the degree. It is different from existing set-based functions such as Jaccard, and has different application domains compared to gram-based methods (e.g. edit distance).

In the context of massive datasets consisting of many long records, if users want to search the records that are semantically similar to a query record, they can integrate the semantical token weights and record-specific degrees into the 2-norm similarity, and model the user preference by placing different degrees in the query record. In this case, the degrees and token weights could be more valuable for both the underlying records and the user interests.

B.3 Extensions Beyond tf-idf

Although our experiments used idf-weighting functions to feed the 2-norm similarity, all our optimization methods do not depend on such special weights. For use-defined token weights, one can define a token order and integrate the weights of processed tokens into the entries on the inverted lists.

In the analysis of inverted index, we suppose that each degree $f(t, R)$ is an integer to denote the frequency of token t in a record R. In many applications, users want to use functional token degrees, e.g., the logistic token frequency [4]. Thus, the degree mappers that have been given in Section 5.1.1 need to be revised to accommodate these data types, such that the functional degrees can also be used in the proposed strategy. While it is easy to revise the mappers for these domain-specific applications, the topics regarding how to update their stepping ranges for best pruning power needs future work.