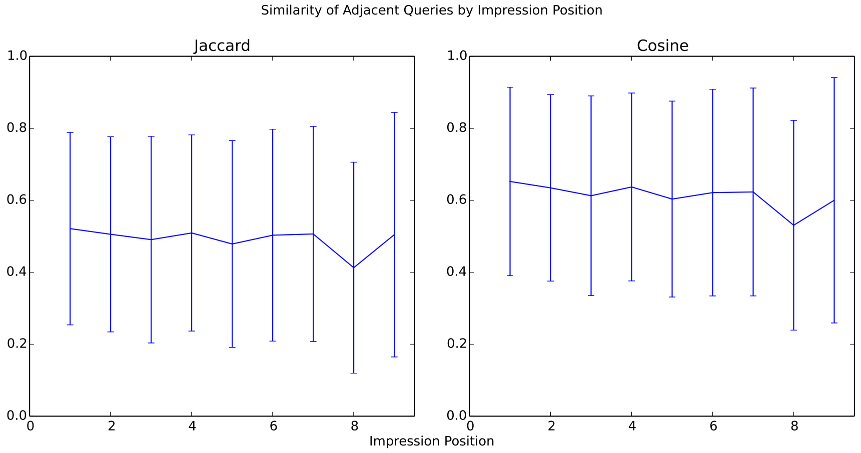

Fig. 2 Average similarity of $q_n \rightarrow q_{n+1}$ pairs for impression positions $n = 1 \ldots 9$

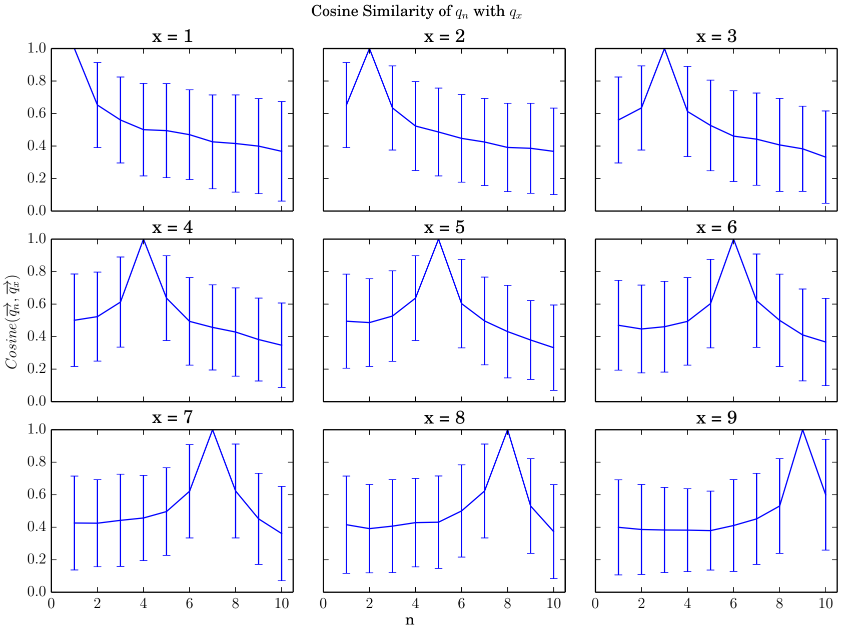

Fig. 3 Cosine similarity of fixed query q_x with every other query q_n in the session for $x = 1 \ldots 9$

5 Term Addition

So far we've found that on average 63% of the terms in query reformulations can be explained by the *retained* or *removed* term actions, leaving 37% of terms unaccounted for. In this section we investigate the *addition* term action which is applicable to added terms a_{n+1} which are terms added from q_n to q_{n+1} i.e. $A_{n+1} = Q_{n+1} \backslash Q_n$. Whereas before we analyzed the similarity of the query reformulation against query terms t_n, in this section we measure the similarity against terms from each of the term sources found in the impression.

When we compare different term sources with a_{n+1} we run into problems caused by term source length. For instance, the *Jaccard* similarity is sensitive to the size of the sets it compares, comparing with a larger set leads to lower similarity, making comparisons between different term sources biased. Additionally, in our studies so far we have been comparing the small number of terms found in queries, where we can consider every term important. Conversely, our term sources can contain hundreds of terms, only a few of which may match the added terms. We counteract these problems in two ways: first we use TFIDF scores (Sparck Jones, 1988) instead of term frequencies in our *Cosine* similarity measure which helps us match on those added terms that are important to the term source. Thus, from this point on any term vectors $\overrightarrow{a}$ refer instead to the TFIDF vector. Secondly, we measure BM25 (Robertson and Zaragoza, 2009) (with typical parameter settings $k_1 = 1.2$ and $b = 0.75$) which is designed to find the similarity of queries consisting of few terms against documents with many terms, and is robust to differing document length. When we use these measures, we treat the collection of all instances of that term source as the document collection for IDF and average document length, for example, the collection of all snippets in the dataset when comparing against a snippet term source.

5.1 Snippet Analysis

We start by considering the snippets found in an impression. A query q_n may have up to M ranked snippets $s_n(k)$ where s is the snippet and k is its rank $1 \leq k \leq M$. In our dataset we join the snippet title onto the snippet under the assumption that anyone reading the snippet has also read its title.

In our first study we look at the similarity of snippets s_n against added terms a_{n+1} at different rank positions. A natural hypothesis based on eye tracking studies Granka et al (2004) is the concept of rank bias, that search results ranked at the top have a higher chance of being observed, thus, they should be more similar to terms added to the next query than lower ranked, potentially unobserved snippets.

In Table 4 we average similarity scores for each snippet $s_n(k)$ from rank 1 to rank k in the impression. Under the assumption given by the Examination Hypothesis model (Craswell et al, 2008) that users examine all snippets in order from the top of the search results to the bottom, we average over *all* snippets up until rank k, not just the snippet at that rank. Our results show that across metrics the similarity peaks at rank positions 2 and 3 before dropping with each rank. The similar lengths of snippets at each rank allows us to rule out a term source length bias. Curiously, the highest ranked snippet on its own does not have the highest similarity to added terms. The implication

Table 4 Average similarity scores between added terms a_{n+1} and snippets s_n up to rank k in an impression. For example, if $k = 3$, then the score is the average over $s_n(1), s_n(2)$ and $s_n(3)$. Maximum values for each similarity measure are in bold.

	k				
	1	2	3	4	5
$Jaccard(A_{n+1}, S_n(k))$	0.00531	**0.00536**	0.00529	0.00507	0.00494
$Cosine(\overrightarrow{a_{n+1}}, \overrightarrow{s_n(k)})$	0.0184	**0.0197**	0.0195	0.0187	0.0185
$BM25(a_{n+1}, s_n(k))$	0.704	0.756	**0.758**	0.737	0.728
# terms in $s_n(k)$	48.3	48.8	49.9	50.2	50.3

Table 5 Average similarity scores between added terms a_{n+1} and snippets s_n up to and around rank LC in an impression, as well as all snippets. Maximum values for each similarity measure are in bold.

	k				
	LC − 1	LC	LC + 1	LC + 2	M
$Jaccard(A_{n+1}, S_n(k))$	0.00440	0.00446	0.00450	0.00458	**0.00465**
$Cosine(\overrightarrow{a_{n+1}}, \overrightarrow{s_n(k)})$	0.0167	0.0171	0.0172	0.0174	**0.0175**
$BM25(a_{n+1}, s_n(k))$	0.656	0.671	0.676	0.682	**0.688**
# terms in $s_n(k)$	51.0	51.0	51.0	50.9	50.4

here is that terms used in query reformulations have a higher chance of being found in the top 2 or 3 ranked snippets and that users don't just consider the top ranked snippet on its own. We note that the examining of the top 2 or 3 search results is consistent with eye tracking observations.

From click model research we can also make the assumption that if we observe a click in an impression, then the user has examined all snippets up until that rank. Let us denote LC as the rank of the Last Click in an impression (that is, the lowest ranked clicked document). In our next study we observe whether similarity change occurs at rank LC and for the snippets ranked above and below it, akin to the 'Click > No-Click Next' strategy and its variants outlined by Joachims et al (2005). If an impression didn't contain a click, then we included all snippets in the impression, our results are in Table 5.

We may have expected a decrease in similarity following rank LC, owing to the hypothesis that a user does not examine documents ranked lower than the last click. In our experiment we find this is not the case, recording a higher similarity score when considering all snippets in an impression rather than just up until the last clicked. A difference between our session search setting and that typically modeled with click models is that in our case, even after a document has been clicked, we know that the user returned to the set of search results in order to issue a reformulation. Conventional click models do not take into account multiple queries in a search session. As such, in our case it is likely that the user continued to examine snippets after the last click, before abandoning the query and issuing a reformulation, leading to our observed results. Also, by comparing these results with those in Table 4 we see that the top ranked 2-3 snippets are still more likely to contain added terms.

These inferences can be observed in our example session in Table 1. For queries q_5 =‘`gun violence us`’ and q_6 =‘`law center to prevent gun violence`’,