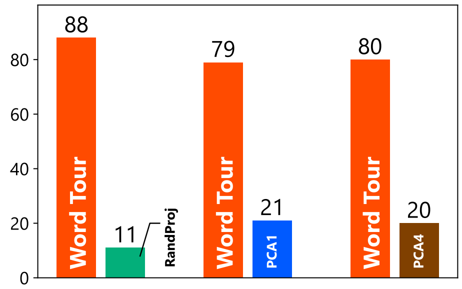

Figure 2: **Results of the user study**. Each bar represents the number of times each method was selected within 100 trials. One trial was not completed in WORDTOUR vs. RandProj, which led to 99 trials in the first comparison.

4.3 Assesment via Crowdsourcing

We conducted a user study at Amazon Mechanical Turk to confirm the effectiveness of WORDTOUR. Specifically, to compare two word ordering $\sigma, \tau \in \mathcal{P}([n])$, we randomly sample a reference word $v \in \mathcal{V}$, retrieve the next words of v in σ and τ, and ask a crowdworker which word is more similar to the reference word v. We repeated this process 100 times for each pair of embeddings. Figure 2 shows the number of times each embedding was selected. This clearly shows that WORDTOUR aligns with human judgment.

4.4 Document Retrieval

In this section, we evaluate the effectiveness of word embeddings in document classification. The most straightforward approach to compare two documents is the bag of words (BoW), which counts common and uncommon words in documents. However, this approach cannot capture the similarities of the words. In 1D embeddings, neighboring words are similar, although they are not exactly matched in BoW. To utilize this knowledge, we use blurred BoW, as shown in Figure 3. Specifically, we put some mass around the words in a document to construct the blurred BoW vector. We employ a Gaussian kernel for the mass amount and use WORDTOUR, RandProj, PCA1, and PCA4 for the orderings. We normalize the BoW and blurred BoW vectors with the L_1 norm and compute the distance between two documents using the L_1 distance of the vectors. The blurred BoW can be computed in $O(wn)$ time, where n denotes the number of words in a document and w is the width of the filter. We used $w = 10$ in the experiments. We also use **word mover's distance (WMD)** (Kusner et al., 2015) as a baseline, which is one of the most popular word-embedding-based distances.

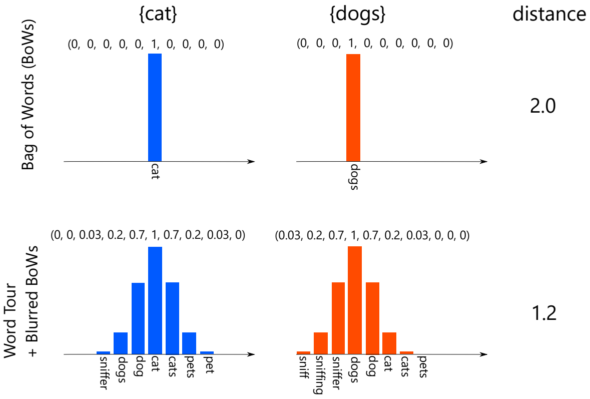

Figure 3: Document comparison by WORDTOUR. This figure illustrates the case in which a document is composed of a single word. When more than one word is in a document, the blurred BoW will be multimodal.

Table 2: **Document classification errors**. *Lower is better*. The time row reports the average time to compare the two documents. WORDTOUR performs the best in the blurred BoW family.

	ohsumed	reuter	20news	amazon	classic
BoW	48.1	5.6	35.4	11.4 ± 0.4	5.1 ± 0.3
Time	39 ns	23 ns	35 ns	21 ns	23 ns
WORDTOUR	**47.2**	**4.6**	**34.1**	**10.1 ± 0.3**	**4.6 ± 0.1**
RandProj	47.9	5.4	35.4	11.3 ± 0.3	5.1 ± 0.3
PCA1	47.8	5.7	35.5	11.4 ± 0.6	5.1 ± 0.3
PCA4	48.1	5.6	35.4	11.6 ± 0.5	5.1 ± 0.4
Time	206 ns	142 ns	312 ns	185 ns	150 ns
WMD	47.5	4.5	30.7	7.6 ± 0.3	4.2 ± 0.3
Time	3.5×10^6 ns	2.2×10^6 ns	5.1×10^6 ns	1.2×10^7 ns	1.9×10^6 ns

We used 300-dimensional GloVe for WMD. WMD requires $O(n^3 + n^2 d)$ computation because of the optimal transport formulation, where n denotes the number of words in a document and d is the number of dimensions of word embeddings. The performance of WMD can be seen as an expensive upper bound of BoW and blurred BoW. We used five datasets: ohsumed (Joachims, 1998), reuter (Sebastiani, 2002), 20news (Lang, 1995), Amazon (Blitzer et al., 2007), and classic (SMART). We remove the duplicated documents following (Sato et al., 2021). The details of the datasets are provided in the Appendix. We evaluated the performance using the k-nearest neighbor error. We used the standard test dataset if it existed (for instance, based on timestamps) and used five random train/test splits for the other datasets[1]. We report the standard deviations for five-fold datasets.

The results are shown in Table 2. Although WORDTOUR is less effective than WMD, it is much faster than WMD and more effective than other 1D embeddings. Recall that the 1D embeddings are designed for low-resource environments, where

[1] The seeds are fixed and reported in the GitHub repository.