Scalable Nearest Neighbor Search for Optimal Transport: 
Supplementary Material

Arturs Backurs¹ ² Yihe Dong³ Piotr Indyk⁴ Ilya Razenshteyn⁵ Tal Wagner⁴

¹Author names are ordered alphabetically. Code available
at https://github.com/ilyaraz/ot_estimators.
²TTIC ³Microsoft ⁴MIT ⁵Microsoft Research. Correspondence
to: Tal Wagner <talw@mit.edu>.

Proceedings of the 37th International Conference on Machine
Learning, Vienna, Austria, PMLR 119, 2020. Copyright 2020 by
the author(s).
A. Proofs

A.1. Flowtree Computation

In this section we prove Lemma 1. We begin by specifying the greedy flow computation algorithm on the tree. Let $h$ denote the height of the tree (for the quadtree the height is $h = O(\log(d))$). Suppose we are given a pair of distributions $\mu, \nu$, each supported on at most $s$ leaves of the tree. For every node $v$ in the tree, let $C_\mu(v)$ denote the set of points in $x \in X$ such that $\mu(x) > 0$ and the tree leaf that contains $x$ is a descendant of $v$. Similarly define $C_\nu(v)$. Note that we only need to consider nodes for which either $C_\mu(v)$ or $C_\nu(v)$ is non-empty, and there are at most $2s h$ such nodes.

The algorithm starts with a zero flow $f$, and processes the nodes in a bottom-up order starting at the leaf. In each node, the unmatched demands collected from its children are matched arbitrarily, and the demands that cannot be matched are passed on to the parent. In mode detail, a node is processed as follows:

1. Collect from the children the list of unmatched $\mu$-demands for the nodes in $C_\mu(v)$ and the list of unmatched $\nu$-demands for the nodes in $C_\nu(v)$. Let $\{\mu_v(x) : x \in C_\mu(v)\}$ denote the unmatched $\mu$-demands and let $\{\nu_v(x) : x \in C_\nu(v)\}$ denote the unmatched $\nu$-demands.

2. While there is a pair $x \in C_\mu(v)$ and $x' \in C_\mu(v)$ with $\mu_v(x) > 0$ and $\nu_v(x') > 0$, let $\eta = \min\{\mu_v(x), \nu_v(x')\}$, and update (i) $f(x,x') + = \eta$, (ii) $\mu_v(x) - = \eta$, (iii) $\nu_v(x') - = \eta$.

3. Now either $\mu_v$ or $\nu_v$ is all-zeros. If the other one is not all-zeros (i.e., there is either remaining unmatched $\mu$-demand or remaining unmatched $\nu$-demand), pass it on to the parent.

A leaf $v$ contains a single point $x \in X$ with either $\mu(x) > 0$ or $\nu(x) > 0$; it simply passes it on to its parent without processing.

It is well known that the above algorithm computes an optimal flow on the tree (with respect to tree distance costs), see, e.g., (Kalantari & Kalantari, 1995). Let us now bound its running time. The processing time per node $v$ in the above algorithm is $O(|C_\mu(v)| + |C_\nu(v)|)$. In every given level in the tree, if $v_1, \ldots, v_k$ are the nodes in that level, then $\{C_\mu(v_1), \ldots, C_\mu(v_k)\}$ is a partition of the support of $\mu$, and $\{C_\nu(v_1), \ldots, C_\nu(v_k)\}$ is a partition of the support of $\nu$. Therefore the total processing time per level is $O(s)$, and since there are $h$ levels, the flow computation time is $O(sh)$. Then we need to compute the Flowtree output $\hat{W}_\ell(\mu, \nu)$. Observe that in the above algorithm, whenever we match demands between a pair $x, x'$, we fully satisfy the unmatched demand of one of them. Therefore the output flow $f$ puts non-zero flow between at most $2s$ pairs. For each such pair we need to compute the Euclidean distance in time $O(d)$, and the overall running time is $O(s(d + h))$.

A.2. Quadtree and Flowtree Analysis

Proof of Theorem 1. Let $x, y \in X$. Let $p_\ell(x, y)$ be the probability that $x, y$ fall into the same cell (hypercube) in level $\ell$ of the quadtree. It satisfies,

$$1 - \frac{||x - y||_1}{2^\ell} \leq p_\ell(x, y) \leq \exp\left(-\frac{||x - y||_1}{2^\ell}\right).$$

(1)

To see this, recall that in level $\ell$ we impose a grid with side length $2^\ell$, shifted at random by an i.i.d. uniform shift in $[0, 2^\ell]$ in each coordinate. The probability that $x, y$ are separated in coordinate $i$ is $2^{-\ell}|x_i - y_i|$, and thus $p_\ell(x, y) = \prod_{i=1}^{d}(1 - 2^{-\ell}|x_i - y_i|)$. The lower bound in Equation (2) follows by a union bound, and the upper bound follows by applying the general estimate $1 - z \leq \exp(-z)$ to each term in the product.

Let $t$ be the tree metric induced on $X$ by the quadtree. Note that for $t(x, y)$ to be at most $O(2^\ell)$, $x, y$ must fall into the same hypercube in level $\ell$. For any $\delta > 0$, we can round $\frac{||x - y||_1}{\log(1/\delta)}$ to its nearest power of 2 and obtain $\ell$ such that $2^\ell = \Theta\left(\frac{||x - y||_1}{\log(1/\delta)}\right)$. It satisfies,

$$\Pr\left[t(x, y) < \frac{O(1)}{\log(1/\delta)}||x - y||_1\right] \leq \delta.$$

By letting $\delta = \Omega(\min\{1/|X|, 1/(s^2 n)\})$, we can take union bound either over all pairwise distances in $X$ (of which there are $\binom{|X|}{2}$), or over all distances between the support of the query $\nu$ and the union of supports of the dataset $\mu_1, \ldots, \mu_n$ (of which there are at most $s^2 n$, if every support has size at most $s$). Then, with probability say 0.995, all those distances are
contracted by at most \(O(\log(\min\{sn, |X|\}))\), i.e.,

\[
t(x, y) \geq \frac{1}{O(\log(1/\delta))} \|x - y\|_1.
\] (2)

On the other hand,

\[
E[t(x, y)] = \sum_{\ell} 2^\ell \cdot (1 - p_\ell(x, y)) \leq \sum_{\ell} 2^\ell \cdot \|x - y\|_1 \leq O(\log(d\Phi)) \cdot \|x - y\|_1.
\]

Let \(\mu^*\) be the true nearest neighbor of \(\nu\) in \(\mu_1, \ldots, \mu_n\). Let \(f_{\mu^*, \nu}^*\) be the optimal flow between them. Then by the above,

\[
E \left[ \sum_{(x, y) \in X \times X} f_{\mu^*, \nu}^*(x, y) t(x, y) \right] \leq O(\log(d\Phi)) \sum_{(x, y) \in X \times X} f_{\mu^*, \nu}^*(x, y) \|x - y\|_1.
\]

By Markov, with probability say 0.995,

\[
\sum_{(x, y) \in X \times X} f_{\mu^*, \nu}^*(x, y) \cdot t(x, y) \leq O(\log(d\Phi)) \sum_{(x, y) \in X \times X} f_{\mu^*, \nu}^*(x, y) \cdot \|x - y\|_1. \tag{3}
\]

Let \(\mu'\) be the nearest neighbor of \(\nu\) in the dataset according to the quadtree distance. Let \(f_{\mu', \nu}^*\) be the optimal flow between them in the true underlying metric (\(\ell_1\) on \(X\)), and let \(f_{\mu, \nu}^*\) be the optimal flow in the quadtree. Finally let \(W_1\) denote the Wasserstein-1 distance on the quadtree. Then,

\[
W_1(\mu', \nu) = \sum_{(x, y) \in X \times X} f_{\mu', \nu}^*(x, y) \cdot \|x - y\|_1
\]

\[
\leq \sum_{(x, y) \in X \times X} f_{\mu', \nu}^* \cdot \|x - y\|_1 \quad \text{\(f_{\mu^*, \nu}^*\) is optimal for \(\|\cdot\|_1\)}
\]

\[
\leq O(\log(\min\{sn, |X|\})) \sum_{(x, y) \in X \times X} f_{\mu', \nu}^* \cdot t(x, y)
\]

\[
= O(\log(\min\{sn, |X|\})) \cdot W_1(\mu', \nu) \quad \text{\text{definition of } } W_1
\]

\[
\leq O(\log(\min\{sn, |X|\})) \cdot W_1(\mu', \nu) \quad \text{\(\mu'\) is the nearest neighbor in } W_1
\]

\[
= O(\log(\min\{sn, |X|\})) \sum_{(x, y) \in X \times X} f_{\mu^*, \nu}^* \cdot t(x, y)
\]

\[
\leq O(\log(\min\{sn, |X|\})) \sum_{(x, y) \in X \times X} f_{\mu^*, \nu}^* \cdot \|x - y\|_1 \quad \text{\text{definition of } } W_1
\]

\[
\leq O(\log(\min\{sn, |X|\}) \log(d\Phi)) \sum_{(x, y) \in X \times X} f_{\mu^*, \nu}^* \cdot \|x - y\|_1 \quad \text{\(f_{\mu^*, \nu}^*\) is optimal for } t(\cdot, \cdot)
\]

\[
= O(\log(\min\{sn, |X|\}) \log(d\Phi)) \cdot W_1(\mu^*, \nu),
\]

so \(\mu'\) is a \(O(\log(\min\{sn, |X|\}) \log(d\Phi))\)-approximate nearest neighbor. \(\square\)

**Proof of Theorem 2.** It suffices to prove the claim for \(s = 1\) (i.e., the standard \(\ell_1\)-distance). Let \(d > 0\) be an even integer. Consider the \(d\)-dimensional hypercube. Our query point is the origin. The true nearest neighbor is \(c_1\) (standard basis vector). The other data points are the hypercube nodes whose hamming weight is exactly \(d/2\). The number of such points is \(\Theta(2^d/\sqrt{d})\), and this is our \(n\).

Consider imposing the grid with cell side 2 on the hypercube. The probability that 0 and 1 are uncut in a given axis is exactly 1/2, and since the shifts in different axes are independent, the number of uncut axes is distributed as \(Bin(d, 1/2)\).
Thus with probability $1/2$ there are at least $d/2$ uncut dimensions. If this happens, we have a data point hashed into the same grid cell as the origin (to get such data point, put 1 in any $d/2$ uncut dimensions and 0 in the rest), so its quadtree distance from the origin is 1. On the other hand, the distance of the origin to its true nearest neighbor $e_1$ is at least 1, since they will necessarily be separated in the next level (when the grid cells have side 1). Thus the quadtree cannot tell between the true nearest neighbor and the one at distance $d/2$, and we get the lower bound $c \geq d/2$. Since $n = \Theta(2^d/\sqrt{d})$, we have $d/2 = \Omega(\log n)$ as desired.

$\square$

**Proof of Theorem 3.** The proof is the same as for Theorem 1, except that in eq. (3), we take a union bound only over the $s^2$ distances between the supports of $\nu$ and $\mu^*$ (the query and its true nearest neighbor). Thus each distance between $\mu^*$ and $\nu$ is contracted by at most $O(\log s)$.

Let $W_F$ denote the Flowtree distance estimate of $W_1$. Let $\mu'$ be the nearest neighbor of $\nu$ in the Flowtree distance. With the same notation in the proof of Theorem 1,

$$W_1(\mu', \nu) = \sum_{(x,y) \in X \times X} f_{\mu',\nu}^*(x, y) \cdot \|x - y\|_1$$

$$\leq \sum_{(x,y) \in X \times X} f_{\mu',\nu}^*(x, y) \cdot \|x - y\|_1$$

$f_{\mu',\nu}^*$ is optimal for $\|\cdot\|_1$ Flowtree definition

$$= W_F(\mu', \nu)$$

$\mu'$ is nearest in Flowtree distance

$$\leq W_F(\mu^*, \nu)$$

$$= \sum_{(x,y) \in X \times X} f_{\mu^*,\nu}^*(x, y) \cdot \|x - y\|_1$$

as needed. Note that the difference from the proof of Theorem 1 is that we only needed the contraction bound (eq. (3)) for distances between $\mu^*$ and $\nu$.

$\square$

**Proof of Theorem 4.** We set $\varepsilon = 1/\log s$. Let $t'(x, y)$ denote the quadtree distance where the weight corresponding to a cell $v$ in level $\ell(v)$ is $2^{\ell(v)(1-\varepsilon)}$ instead of $2^{\ell(v)}$. Let $f_{\mu',\nu}$ be the optimal flow in the quadtree defined by weights $t'$.

Let $\delta = c/s^2$ where $c > 0$ is a sufficiently small constant. For a every $x, y$, let $\ell_{xy}$ be the largest integer such that

$$2^{\ell_{xy}} \leq \frac{\|x - y\|_1}{(\log(1/\delta))^{1/(1-\varepsilon)}}.$$

The probability that $x, y$ are separated (i.e., they are in different quadtree cells) in level $\ell_{xy}$ is

$$1 - p_{\ell_{xy}}(x,y) \geq 1 - \exp \left( - \frac{\|x - y\|_1}{2^{\ell_{xy}}} \right) \geq 1 - \frac{\delta}{1 - \varepsilon}.$$

By the setting of $\delta$, we can take a union bound over all $x \in \text{support}(\mu^*)$ and $y \in \text{support}(\nu)$ and obtain that with say 0.99 probability, simultaneously, every pair $x, y$ is separated at level $\ell_{xy}$. We denote this event by $E_{\text{lower}}$ and suppose it occurs. Then for every $x, y$ we have

$$t'(x, y) \geq 2 \cdot 2^{\ell_{xy}(1-\varepsilon)} \geq 2 \cdot \left( \frac{1}{2} \frac{\|x - y\|_1}{(\log(1/\delta))^{1/(1-\varepsilon)}} \right)^{1-\varepsilon} \geq \frac{\|x - y\|_1^{1-\varepsilon}}{\log(1/\delta)} = \frac{\|x - y\|_1^{1-\varepsilon}}{\Theta(\log s)}.$$
Next we upper-bound the expected tree distance \( t'(x, y) \). (Note that we are not conditioning on \( E_{\text{lower}} \).) Observe that

\[
t'(x, y) = 2 \sum_{\ell=-\infty}^{\infty} 2^{\ell(1-\epsilon)} \cdot 1\{x, y \text{ are separated at level } \ell\}.
\]

Let \( L_{x,y} \) be the largest integer such that \( 2^{L_{x,y}} \leq \|x - y\|_1 \). We break up \( t'(x, y) \) into two terms,

\[
t'_{\text{lower}}(x, y) = 2 \sum_{\ell=-\infty}^{L_{x,y}} 2^{\ell(1-\epsilon)} \cdot 1\{x, y \text{ are separated at level } \ell\},
\]

and

\[
t'_{\text{upper}}(x, y) = 2 \sum_{\ell=L_{x,y}+1}^{\infty} 2^{\ell(1-\epsilon)} \cdot 1\{x, y \text{ are separated at level } \ell\},
\]

thus \( t'(x, y) = t'_{\text{lower}}(x, y) + t'_{\text{upper}}(x, y) \). For \( t'_{\text{lower}}(x, y) \) it is clear that deterministically,

\[
t'_{\text{lower}}(x, y) \leq 2 \sum_{\ell=-\infty}^{L_{x,y}} 2^{\ell(1-\epsilon)} = O\left(2^{L_{x,y}(1-\epsilon)}\right) = O\left(\|x - y\|_1^{1-\epsilon}\right).
\]

For \( t'_{\text{upper}}(x, y) \), we have

\[
\mathbb{E}[t'_{\text{upper}}(x, y)] = 2 \sum_{\ell=L_{x,y}+1}^{\infty} 2^{\ell(1-\epsilon)} p_\ell(x, y)
\]

\[
\leq 2 \sum_{\ell=L_{x,y}}^{\infty} 2^{\ell(1-\epsilon)} \cdot \frac{\|x - y\|_1}{2^{\ell}}
\]

\[
= 2\|x - y\|_1 \sum_{\ell=L_{x,y}}^{\infty} 2^{-\ell \epsilon}
\]

\[
= 2\|x - y\|_1 \cdot \frac{2^{-L_{x,y} \epsilon}}{1 - 2^{-\epsilon}}
\]

\[
\leq O(\log s) \cdot \|x - y\|_1^{1-\epsilon},
\]

where in the final bound we have used that \( 2^{L_{x,y}} = \Theta(\|x - y\|_1) \) and \( 1 - 2^{-\epsilon} = \Theta(\epsilon) = \Theta(\log s) \). Together,

\[
\mathbb{E}[t'(x, y)] = \mathbb{E}[t'_{\text{lower}}(x, y) + t'_{\text{upper}}(x, y)] \leq \Theta(\log s) \cdot \|x - y\|_1^{1-\epsilon}.
\]

(4)

Now we are ready to show the \( O(\log^2 s) \) upper bound on the approximation factor. Below we will use the fact that every weight \( f_{\mu', \nu}(x, y) \) in the flow is of the form \( i/(s's'') \) for some integer \( 0 \leq i \leq s's'' \). This follows from the assumption that each element in the support of every measure is an integer multiple of \( 1/s' \) or of \( 1/s'' \) for some \( 1 \leq s', s'' \leq s \).

\[
W_1(\mu', \nu) = \sum_{(x,y) \in X \times X} f^*_{\mu', \nu}(x, y) \cdot \|x - y\|_1
\]

\[
\leq \sum_{(x,y) \in X \times X} f_{\mu, \nu}(x, y) \cdot \|x - y\|_1
\]

\[
= W_F(\mu', \nu)
\]

\[
\leq W_F(\mu^*, \nu)
\]

\[
= \sum_{(x,y) \in X \times X} f_{\mu^*, \nu}(x, y) \cdot \|x - y\|_1
\]

\( f_{\mu', \nu} \) is optimal for \( \|\cdot\|_1 \) \hspace{10cm} \( f_{\mu^*, \nu} \) is nearest to \( \nu \) in Flowtree distance

Flowtree definition

\( \mu' \) is nearest to \( \nu \) in Flowtree distance

Flowtree definition
Sinkhorn has a regularization parameter. In those experiments too, the best results are achieved with either 1 or 3 iterations to ACT-1, Sinkhorn-1 and Sinkhorn-3. We also remark that in the pipeline experiments, we have evaluated Sinkhorn with used in Section 4 yields comparable accuracy with a slower running time. Therefore in Section 4 we restrict our evaluation on each of our datasets.

Both ACT and Sinkhorn are iterative algorithms, and the number of iterations is a parameter to set. In order for Quadtree to correctly identify \(\mu\) as the nearest neighbor of \(\nu\), every \(H_k\) must not contain any additional points from \(X\). Otherwise, if say \(H_1\) contains a point \(x' \neq x_1\), the \(W_1\) distance on the quadtree from \(\nu\) to \(\mu_i\) is equal to its distance to the uniform distribution over \(\{x', x_2, \ldots, x_s\}\). Since the points in \(X\) are chosen uniformly i.i.d. over \(S^{d-1}\), the probability of the above event, and thus the success probability of Quadtree, is upper bounded by \(\mathbb{E}[(1 - V)^{N-s}]\), where \(V = \text{volume}(\bigcup_{k=1}^s H_k \cap S^{d-1})\). This \(V\) is a random variable whose distribution depends only on \(d, s, \epsilon\), and is independent of \(N\). Thus the success probability decays exponentially with \(N\).

Flowtree. On the other hand, suppose that each \(H_k\) contains no other points from \(\{x_1, \ldots, x_s\}\) other than \(x_k\) (but is allowed to contain any other points from \(X\)). This event guarantees that the optimal flow on the tree between \(\mu_i\) and \(\nu\) is the planted perfect matching, i.e., the true optimal flow, and thus the estimated Flowtree distance between them \textit{equals} \(W_1(\mu_i, \nu)\). This guarantees that Flowtree recovers the planted nearest neighbor, and this event depends only on \(d, s, \epsilon\), and is independent of \(N\).

B. Additional Experiments

B.1. Additional Sinkhorn and ACT Experiments

Number of iterations. Both ACT and Sinkhorn are iterative algorithms, and the number of iterations is a parameter to set. Our main experiments use ACT with 1 iteration and Sinkhorn with 1 or 3 iterations. The next experiments motivate these choices. Figures 7(a)–(c) depict the accuracy and running time of ACT-1, ACT-7, Sinkhorn-1, Sinkhorn-3 and Sinkhorn-5 on each of our datasets.\(^1\) It can be seen that for both algorithms, increasing the number of iterations beyond the settings used in Section 4 yields comparable accuracy with a slower running time. Therefore in Section 4 we restrict our evaluation to ACT-1, Sinkhorn-1 and Sinkhorn-3. We also remark that in the pipeline experiments, we have evaluated Sinkhorn with up to 9 iterations. In those experiments too, the best results are achieved with either 1 or 3 iterations.

Sinkhorn regularization parameter. Sinkhorn has a regularization parameter \(\lambda\) that needs to be tuned per dataset. We set \(\lambda = \eta \cdot M\), where \(M\) is the maximum value in the cost matrix (of the currently evaluated pair of distributions), and

\(^1\)ACT-1 and ACT-7 are the settings reported in (Atasu & Mittelholzer, 2019).
tune $\eta$. In all of our three datasets the optimal setting is $\eta = 30$, which is the setting we use in Section 4. As an example, Figure 7(d) depicts the 1-NN accuracy (y-axis) of Sinkhorn-1 per $\eta$ (x-axis).

Figure 1: Additional Sinkhorn and ACT experiments

![Graphs showing 1-NN accuracy for different datasets and regularization values.](image)

(a) 20news dataset  
(b) Amazon dataset  
(c) MNIST dataset  
(d) 1-NN accuracy of Sinkhorn-1 with varying regularization

B.2. Additional Pipeline Results

The next tables summarize the running times and parameters settings of all pipelines considered in our experiments (whereas the main text focuses on pipelines that start with Quadtree, since it is superior as a first step to Mean and Overlap). The listed parameters are the number of output candidates of each step in the pipeline.

In the baseline pipelines, parameters are tuned to achieve optimal performance (i.e., minimize the running time while attaining the recall goal on at least 90% of the queries). The details of the tuning procedure is as follows. For all pipelines we use the same random subset of 300 queries for tuning. Suppose the pipeline has $\ell$ algorithms. For $i = 1, \ldots, \ell$, let $c_i$ the output number of candidates of the $i$th algorithm in the pipeline. Note that $c_\ell$ always equals either 1 or 5, according to the recall goal of the pipeline, so we need to set $c_1, \ldots, c_{\ell-1}$. Let $p_1$ be the recall@1 accuracy of the first algorithm in the pipeline. Namely, $p_1$ is the fraction of queries such that the top-ranked $c_1$ candidates by the first algorithm contain the true nearest neighbor. We calculate 10 possible values of $c_1$, corresponding to $p_1 \in \{0.9, 0.91, \ldots, 0.99\}$. We optimize the pipeline by a full grid search over those values of $c_1$ and all possible values of $c_2, \ldots, c_{\ell-1}$.

When introducing Flowtree into a pipeline as an intermediate method, we do not re-optimize the parameters, but rather set its output number of candidates to the maximum between 10 and twice the output number of candidates of the subsequent algorithm in the pipeline. Re-optimizing the parameters could possibly improve results.
### Table 1: Recall@1, no Flowtree.

| Pipeline methods                  | Candidates | Time |
|-----------------------------------|------------|------|
| Mean, Sinkhorn-1, Exact           | 1476, 11, 1| 0.543|
| Mean, Sinkhorn-3, Exact           | 1476, 5, 1 | 0.598|
| Mean, R-WMD, Exact                | 1850, 28, 1| 0.428|
| Mean, ACT-1, Exact                | 1677, 14, 1| 0.420|
| Overlap, Sinkhorn-1, Exact        | 391, 6, 1  | 0.610|
| Overlap, Sinkhorn-3, Exact        | 391, 5, 1  | 0.691|
| Overlap, R-WMD, Exact             | 576, 14, 1 | 0.367|
| Overlap, ACT-1, Exact             | 434, 10, 1 | 0.429|
| Quadtree, Sinkhorn-1, Exact       | 295, 5, 1  | 0.250|
| Quadtree, Sinkhorn-3, Exact       | 227, 3, 1  | 0.248|
| Quadtree, R-WMD, Exact            | 424, 12, 1 | 0.221|
| Quadtree, ACT-1, Exact            | 424, 8, 1  | 0.236|

### Table 2: Recall@1, with Flowtree.

| Pipeline methods                  | Candidates | Time |
|-----------------------------------|------------|------|
| Mean, Flowtree, Sinkhorn-1, Exact | 1850, 10, 5, 1 | 0.089|
| Mean, Flowtree, Sinkhorn-3, Exact | 1677, 10, 4, 1 | 0.077|
| Mean, Flowtree, R-WMD, Exact      | 2128, 48, 24, 1 | 0.242|
| Mean, Flowtree, ACT-1, Exact      | 2128, 20, 10, 1 | 0.138|
| Overlap, Flowtree, Sinkhorn-1, Exact | 489, 10, 5, 1 | 0.087|
| Overlap, Flowtree, Sinkhorn-3, Exact | 576, 10, 3, 1 | 0.076|
| Overlap, Flowtree, R-WMD, Exact   | 576, 28, 14, 1 | 0.173|
| Overlap, Flowtree, ACT-1, Exact   | 576, 16, 8, 1 | 0.119|
| Quadtree, Flowtree, Sinkhorn-1, Exact | 424, 10, 5, 1 | 0.074|
| Quadtree, Flowtree, Sinkhorn-3, Exact | 424, 10, 3, 1 | 0.059|
| Quadtree, Flowtree, R-WMD, Exact  | 424, 22, 11, 1 | 0.129|
| Quadtree, Flowtree, ACT-1, Exact  | 424, 16, 8, 1 | 0.104|
| Mean, Flowtree, Exact             | 1850, 9, 1  | 0.105|
| Overlap, Flowtree, Exact          | 489, 9, 1   | 0.100|
| Quadtree, Flowtree, Exact         | 424, 9, 1   | 0.092|

### Table 3: Recall@5, no Flowtree.

| Pipeline methods                  | Candidates | Time |
|-----------------------------------|------------|------|
| Mean, Sinkhorn-1                  | 1476, 5    | 0.464|
| Mean, Sinkhorn-3                  | 1476, 5    | 0.549|
| Mean, R-WMD, Exact                | 1850, 28, 5| 0.426|
| Mean, ACT-1, Exact                | 1677, 14, 5| 0.423|
| Overlap, Sinkhorn-1               | 391, 5     | 0.560|
| Overlap, Sinkhorn-3               | 391, 5     | 0.650|
| Overlap, R-WMD, Exact             | 576, 14, 5 | 0.368|
| Overlap, ACT-1, Exact             | 434, 10, 5 | 0.428|
| Quadtree, Sinkhorn-1              | 295, 5     | 0.222|
| Quadtree, Sinkhorn-3              | 227, 5     | 0.200|
| Quadtree, R-WMD, Exact            | 424, 11, 5 | 0.216|
| Quadtree, ACT-1, Exact            | 424, 7, 5  | 0.222|
| Pipeline methods | Candidates | Time  |
|------------------|------------|-------|
| Mean, Flowtree, Sinkhorn-1 | 1850, 10, 5 | 0.046 |
| Mean, Flowtree, Sinkhorn-3 | 1476, 10, 5 | 0.043 |
| Mean, Flowtree, R-WMD, Exact | 2128, 48, 24, 5 | 0.237 |
| Mean, Flowtree, ACT-1 | 2128, 10, 5 | 0.048 |
| Overlap, Flowtree, Sinkhorn-1 | 391, 10, 5 | 0.042 |
| Overlap, Flowtree, Sinkhorn-3 | 391, 10, 5 | 0.044 |
| Overlap, Flowtree, R-WMD, Exact | 576, 28, 14, 5 | 0.173 |
| Overlap, Flowtree, ACT-1 | 576, 10, 5 | 0.046 |
| Quadtree, Flowtree, Sinkhorn-1 | 424, 10, 5 | 0.033 |
| Quadtree, Flowtree, Sinkhorn-3 | 424, 10, 5 | 0.034 |
| Quadtree, Flowtree, ACT-1 | 424, 10, 5 | 0.029 |
| Mean, Flowtree | 2128, 5 | 0.043 |
| Overlap, Flowtree | 576, 5 | 0.039 |
| Quadtree, Flowtree | 645, 5 | 0.027 |
| Quadtree, Flowtree, R-WMD, Exact | 424, 22, 11, 5 | 0.131 |
| Quadtree, Flowtree, ACT-1, Exact | 424, 16, 8, 5 | 0.103 |

Table 4: Recall@5, with Flowtree.