TopSig at the SIGIR'eCom 2018 Rakuten Data Challenge

Timothy Chappell
School of Electrical Engineering and Computer Science, Queensland University of Technology
Brisbane, Queensland
timothy.chappell@qut.edu.au

Shlomo Geva
School of Electrical Engineering and Computer Science, Queensland University of Technology
Brisbane, Queensland
s.geva@qut.edu.au

Lawrence Buckingham
School of Electrical Engineering and Computer Science, Queensland University of Technology
Brisbane, Queensland
l.buckingham@qut.edu.au

ABSTRACT

For the SIGIR 2018 eCom workshop Rakuden Data Challenge we present an approach utilising random indexing to create toplogical signatures (TopSig) for the purposes of categorising the product names in the provided data sets. This paper describes the approach used to accomplish this.

CCS CONCEPTS

• **Information systems → Clustering and classification**;

KEYWORDS

Signatures, hamming distance

ACM Reference Format:
Timothy Chappell, Shlomo Geva, and Lawrence Buckingham. 2018. TopSig at the SIGIR'eCom 2018 Rakuten Data Challenge. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2018).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Traditional text search engines are based on the bag-of-words approach, and are almost invariably implemented with inverted files [7]. There are however some information retrieval tasks that are better handled by alternative techniques, such as locality sensitive hashing (LSH). These are often referred to as approximate nearest neighbour approaches and are common in the image or speech domains, where objects are typically represented by numerical quantities and where text based approaches are not naturally applicable. With LSH, the data can be handled in a more convenient way by projecting from the original feature space, where the natural representation of an object is inherently high dimensional with variable length, into a space with fixed dimensionality. It then becomes possible to use similarity between object representations as a proxy for calculation of original representation object similarity. The fixed dimensionality of the representation space combined with favourable properties of the metric in that space permit the implementation of highly efficient search and comparison algorithms.

SIGIR 2018, July 2018, Ann Arbor, Michigan USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Unlike text search engines, LSH based search engines compare entire object representations, rather than look for individual features within objects (key words). In this project we aim to use the LSH approach for searching the Rakuten items text collection, by taking the entire object comparison approach, treating each item description as an object. We then project all items into a lower dimensional metric space, where it can be efficiently processed.

This experiment is concerned with classification rather than retrieval. We use a conceptually simple approach:

- We first search the collection with the new product title and identify its nearest neighbours in the reference data set.
- We then apply a simple post-search re-ranking of the top-k nearest neighbours, and assign to the new item the category of the highest ranked item from the top-k nearest.

In this paper we apply the Topological Signature (TopSig) approach to this challenge to produce the initial top-k list.

2 TOPOLOGICAL SIGNATURES

The concept of performing similarity computations within a model in which documents exist as vectors is an old one [5], but still considered a fundamental of information retrieval theory today. The naturally high dimensionality of text of any level of complexity poses certain computational challenges; therefore it is desirable to reduce this dimensionality where possible. Singular value decomposition has been used successfully for this purpose [2], however it has also been shown that random indexing [4] can achieve similar results with far less preprocessing time.

For the purposes of this research, we have made use of our open-source TopSig tool to generate dimensionality-reduced signatures from the product titles in the provided data sets and to search these signatures. The signature generation approach used by TopSig has been shown to be effective at ad-hoc document retrieval tasks [3], biological sequence clustering tasks [1] and image retrieval tasks [6] among others. The approach can be regarded as a variation on the LSH approach to approximate nearest neighbour search, and is further described in the following section.

3 SIGNATURE GENERATION

The signature generation process involves transforming the plain text product titles into fixed length binary strings (signatures). Random indexing of text starts with the assignment of a fixed length random signature to each term in the collection. These term signatures can take the form of ternary vectors having components in the set $[+1, 0, -1]$, or they can be normally-distributed random real valued unit vectors. To index a document, the signatures of all terms appearing in the document are added together, and the

```
for all document ∈ collection do
  for all term ∈ document do
    document vector ← document vector + hash(term)
  end for
  for all component ∈ document vector do
    if document vector[component] ≤ 0 then
      document signature[component] ← 0
    else
      document signature[component] ← 1
    end if
  end for
end for
```

Figure 1: Pseudo-code algorithm for generating signatures

resulting vector is then squashed to yield a binary vector by taking the sign bit of each vector component. The randomly generated term signatures play a role analogous to the basis vectors of a projection into the fixed-dimension metric space. When combined they yield binary vectors with effectively random components which nonetheless reflect the term content of the originating document. This projection preserves topology in the sense that documents containing similar terms tend to have similar signatures. This is a special case of *locality-sensitive hashing*. Similarity calculations between signatures are based on the Hamming distance (number of differing bits between the signatures). By encoding and processing signatures with bitwise operations distance calculations can be extremely fast.

TopSig is used to generate signatures from the product titles provided with this challenge through the following process:

- The product titles are filtered, with non-alphabetical characters replaced with spaces.
- The titles are tokenised along space boundaries into individual terms.
- The individual terms are checked against a general term 'stop list' consisting of common sentence particles ('a', 'the' etc.) and terms matching terms in the stop list are discarded.
- The terms are stemmed- words ending in 's', 'es' or 'ies' have these endings removed.
- The terms are case-folded to ensure that the same signatures are generated irrespective of the capitalisation of the terms.
- Finally, the terms are hashed to signatures through random projection, then summed together to form a signature for the entire title, in the process depicted in Figure 1.

The use of pseudo-random hashing in the signature generation process creates signatures in which two completely unrelated titles will tend to have approximately 50% of bits matching when their signatures are compared. The presence of co-occurring terms will increase the number of matching bits, while the presence of *only* co-occurrent terms will result in two precisely-matching signatures. In this way, the number of term co-occurrences between two titles can be approximated by comparing their signatures. It is this relationship that forms the basis of TopSig's signature search approach.

```
for all word ∈ query do
  query vector ← query vector + hash(word)
end for
for all component ∈ query vector do
  if query vector[component] ≤ 0 then
    query sig[component] ← 0
  else
    query sig[component] ← 1
  end if
end for
for all document sig ∈ collection do
  score[document] ← distance(document sig, query sig)
end for
```

Figure 2: Pseudo-code algorithm for exhaustively searching signatures (object⟷object comparison)

4 SIGNATURE SEARCHING

Once the signatures for a collection have been created, the task of searching them is simple. When the collections are small, such as in the case of the Rakuten Data Challenge, the approach that leads to the lowest potential for error is to leverage the high speed at which Hamming distances between signatures can be computed and to exhaustively search the database collection for the query signature. The simplest manifestation of this approach is shown in Figure 2, in which the query is first transformed into a signature, then the resulting signature is compared against every other signature in the database, accumulating a score for each one. These scores (lower means closer to the query signature) can then be used, depending on the needs of the particular domain, to determine the closest results and handle them in whatever way is needed.

One innovation introduced by the TopSig approach that produces more stable results when searching is to take advantage of the fact that we have access to the query at the time of searching, which means we can produce a mask that excludes the bits that are not part of the query signature (either positively or negatively).

The process for this is as follows:

- During searching, when the signature is being generated from the query terms, a second shadow signature (the *mask signature*) is generated as well.
- Whenever a positive or negative value is stored in a term signature, the mask signature receives a 1 in that position.
- Then, as the database is being exhaustively searched, during the Hamming distance calculation both the query and database signatures are masked against the mask signature with a bitwise AND operation, causing the bits that are not part of the mask to be excluded from the final result.

This refined approach is described in Figure 3. Note that, for this to feature any improvement over traditional searching, the term vectors produced by hashing need to be relatively sparse, with many components having a value of 0 and therefore not contributing to the overall signature.

Due to the way signatures are generated, bits that are not featured in the query signature will have a 50% chance of agreeing with the database signatures. This results in a binomial distribution of noise centred on half the number of unset bits affecting the

```
for all word ∈ query do
    query vector ← query vector + hash(word)
end for
for all component ∈ query vector do
    if query vector[component] ≤ 0 then
        query sig[component] ← 0
    else
        query sig[component] ← 1
    end if
    if query vector[component] ≠ 0 then
        mask sig[component] ← 0
    else
        mask sig[component] ← 1
    end if
end for
for all document sig ∈ collection do
    score[document] ← distance(document sig ∧ mask sig, query sig ∧ mask sig)
end for
```

Figure 3: Pseudo-code algorithm for exhaustively searching signatures (masked query⟷object comparison)

Hamming distance of all comparisons, which can be a significant distortion if the query signature contains far fewer terms than the database signature. Masking mitigates this effect by removing these bits from the Hamming distance calculation entirely.

5 DATA CHALLENGE

The Rakuten data challenge provides a data set of 1,000,000 product names from https://www.rakuten.com/, divided into a training set of 800,000 products with category labels intact and a test set of 200,000 products with labels omitted. These labels are presented as a sequence of numbers, where each number represents a category at a particular level on the Rakuten.com website. For instance, the `1208>546>4262>2775` category refers to the Food & Beverage → Beverages → Wine → United States category of products:

- 1208 → Food & Beverage
- 546 → Beverages
- 4262 → Wine
- 2775 → United States

The goal in this challenge is to predict the labels in the test set.

While the labels do form a hierarchy in this fashion, we found that the supercategories in many cases covered too broad a range of items, making predicting them quite difficult. As a result, our attempts at hierarchically categorising products by predicting the category at each level, then searching within the subset of results within that category to predict the next level was less accurate than simply predicting the label directly. As a result, the approach we describe in the following section does not make use of the hierarchical structure of the categories, but instead predicts the entire chain.

```
MaxHD ← results[k − 1].distance
for all result ∈ results do
    LQ ← | query.terms |
    LR ← | result.terms |
    LI ← 2^|query.terms∧result.terms|
    L ← (LI/LQ) min{LQ/LR, LR/LQ}
    HD ← result.distance
    H ← e^(1.0 − (HD−MaxHD)/128)
    score[result.class] ← score[result.class] + L × H
end for
```

Figure 4: Pseudo-code algorithm for the scoring approach applied to the top-k to generate scores for each class

6 APPLYING TOPSIG TO THE CHALLENGE

To produce our attempt at this task, we first cleaned the data, replacing all non-alphabetical characters with spaces, then used TopSig to generate 2048-bit binary signatures for every topic in the training set. Then, to predict the categories of products in the test set, we converted each test title to a 2048-bit signature with the same approach. Following this, we performed an exhaustive search against the training signatures with the test signature using the masked search variant described in Figure 3, computing the Hamming distance between the test signature and each training signature, and recording the closest 10 signatures.

7 POST-PROCESSING

While the initial signature search delivers the correct class in the majority ($\geq$ 77%) of cases, we find that in certain cases the top result does not return the correct class, while the correct class is represented well in the remaining top-k results. To handle this possibility, we include a post-processing step that is applied to re-rank the top-k, pushing well-represented classes to the top if the top result happens to be an outlier.

During post-processing, we iterate over the top-k results and accumulate a score for each class represented in that set based on two factors, L and H.

- L reflects the number of intersecting terms between the query and each result.
- H is derived from the difference between the Hamming distance of the current result and the last result in the top-k.

These two factors are then multiplied together to produce a score, which is accumulated for each class represented in the top-k. The scoring algorithm is illustrated in Figure 4.

Finally, the class with the highest score is predicted as the outcome for this particular search.

8 EVALUATION

Classification accuracy is measured by comparing the predicted category for each product with the actual category. Results are reported in terms of the following metrics:

- precision
- recall
- F_1-score

Among these, the F_1-score is the score ultimately used to compare systems, although all values are reported for competing systems.

9 RESULTS

The approach described in this paper, consisting of a masked signature search followed by the post-processing described in section 7 achieved the following precision, recall and F_1-scores on the testing set in the Rakuten Data Challenge:

- Precision: 0.80
- Recall: 0.80
- F_1-score: 0.80

10 CONCLUSION

We have presented our contribution to the SIGIR'eCom 2018 Rakuten Data Challenge. The results show that, with minimal customisation and only a small amount of post-processing, TopSig's signature generation and search capabilities can be used as an effective classification tool. Even a very basic signature search produces reasonable results initially, with a little bit of post-processing capable of pushing the classification accuracy even higher.

11 SOURCE CODE AND DATA

The TopSig signature generation and search tool we used for this challenge is available from http://www.topsig.org and is licensed under the GNU GPLv3. The data is available from the organisers of the SIGIR'eCom 2018 Rakuten Data Challenge (https://sigir-ecom.github.io/data-task.html).

12 ACKNOWLEDGEMENTS

We would like to acknowledge the computational resources provided by the Queensland University of Technology Science and Engineering Faculty's BigData Lab for the contribution these resources played in our attempt at this challenge.

REFERENCES

[1] Timothy Chappell, Shlomo Geva, and James Hogan. 2017. K-Means Clustering of Biological Sequences. In *Proceedings of the 22nd Australasian Document Computing Symposium*. ACM, 2.

[2] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science* 41, 6 (1990), 391–407.

[3] Shlomo Geva and Christopher M De Vries. 2011. TopSig: topology preserving document signatures. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*. ACM, 333–338.

[4] M. Sahlgren. 2005. An introduction to random indexing. In *Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE*, Vol. 5.

[5] Gerard Salton, Anita Wong, and Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Commun. ACM* 18, 11 (1975), 613–620.

[6] Nanayakkara Wasam Uluwitige, Dinesha Chathurani, Shlomo Geva, Timothy Chappell, and Vinod Chandran. 2016. Efficient content-based image retrieval based on multi-feature fusion. (2016).

[7] J. Zobel, A. Moffat, and K. Ramamohanarao. 1998. Inverted files versus signature files for text indexing. *ACM Transactions on Database Systems (TODS)* 23, 4 (1998), 453–490.