Information Object Naming on embedding-based approach

Alexey Kolosov
Moscow State University
akolosov@cs.msu.ru

Ruslan Smeliansky
Moscow State University
smel@cs.msu.ru

ABSTRACT

The Information Object Naming problem within Information Centric Network (ICN) concept is considered. As Information Objects (IO) text documents are under consideration. In ICN concept there two form of the name for the same IO – Information Object Name (ION) and Name Identifier (NID). The first one is used for user manipulation, the second one – for IO name representation within ICN. The article shows how to construct a coordinate ION space and proposes a method to identify a specific ION in that space. A new, distinctive feature of the proposed method is that it provides semantically close IOs with close ION in coordinate space. The proposed method is based on embedding technics.

ACM Reference Format:
Alexey Kolosov and Ruslan Smeliansky. 2020. Information Object Naming on embedding-based approach. In *Proceedings of CoNEXT '20*. ACM, New York, NY, USA, 5 pages. https://doi.org/TBA

1 INTRODUCTION

In the ICN, access to the IO is by name, not the address of the device containing the IO. In such organized networks, access to IO does not require knowledge of its location. IOs may have duplicates placed on the network so as to increase access efficiency, for example, by the criterion of demand. Therefore, there is a logical separation of the level of presentation of data and the level of access to data.

IO names have two forms: ION and NID. ION is used to represent the IO of the user, and NID is used to represent the IO in the system. In other words, ION is interpreted by human, and NID is interpreted by a computer. A situation is possible in which the ION and NID representations are the same, but the interpretations are different.

In ICN, two subsystems are associated with names: the Naming System (NS), which is responsible for mapping a set of IOs to a set of names, and the Denaming System (DS), which is responsible for mapping a set of names to a set of IOs. The article formulates the requirements for the NS subsystem as well as possible ways of their implementation for the case of text objects.

The relevance of the problem is caused by the fact that new IOs are constantly generated on the Internet, and in ICN, the effectiveness of access to IOs is associated with the effectiveness of NS organization, which is determined by the requirements for it.

The structure of the work is as follows: Section 2 is devoted to the history of the emergence of ICN, a discussion of well-known approaches to organizing the architecture of ICN. Section 3 discusses the tasks for constructing the NS subsystem and formulates methods for solving them. Section 4 addresses further research.

2 RELATED WORK

In 2009, Van Jacobson introduced the concept of information-oriented networks (ICN) [1], which declared the transition from the host-centric paradigm to information-centric. The main postulate was access to information objects (IO) by their names, and not the addresses of devices containing them. Thus, in the ICN it is possible to access the IO without knowing its location. This approach allows us to separate the levels of data representation and access to data, which in turn leads to the fact that data can move across the network to places where it is most in demand.

In subsequent years, several options for architectures ICN were proposed: CCN, NDN [2], DONA [3]. A common feature of all these architectures is that access to the object is based on prefix routing.

CCN and its NDN successor used an approach where each node contained 3 tables: Pending Interest Table (PIT), Forwarding Information Base (FIB), Content Store (CS). Access to the EUT is done by sending an Interest Packet, the response to which is a Data Packet containing the requested EUT or its part. The names in the CCN and NDN are hierarchical and are contained in the Interest Packet. This approach allows to

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CoNEXT '20, December 4–7, 2020, Greece

© 2020 Association for Computing Machinery.

ACM ISBN TBA. . . $TBA
https://doi.org/TBA

use prefix routing to determine the location of the IO. When searching for an IO, Interest is stored in the PIT, and the next hop is determined using the FIB. After the IO is detected, it is transmitted along the route back to the path in which it was found. In DONA, the same approach was implemented as in DNS, only devices were replaced with IO.

The host-centric approach uses a combined routing table to navigate in the name space and devices, as devices are named. When scaling the system, the time it takes for changes in the routing tables, in the case of routing by prefixes, when adding or moving the IO, to propagate through a distributed system, will increase — i.e. convergence rate will decrease.

In data-centric, it is possible to split tables for routing in the name space and devices, since the data is named. In NDN and DONA, this separation is not completely made, since in the case of prefix routing, there remains a connection between the parts of the names and the next hop in the physical network, in the FIB. Therefore, when scaling a distributed system, the convergence rate will also decrease, although not so significantly.

In the proposed approach, the following separation is present: there is a separate table for routing in the name space, Name Space Table (NST) and a separate table for routing between devices, Device Space Table (DST), there is also a map between these two tables, Map Space Table (MST). This solves the problem of reducing the convergence rate when scaling a distributed system, since the number of routing rules in NST does not depend on the number of routing rules in DST, vice versa. Thus, when moving the IO, the display between the tables changes, but not the tables themselves.

3 SYSTEM MAIN COMPONENTS

The ICN consists of four components: Naming System (NS), Denaming System (DS), Forwarding System (FS), Storage System (SS). NS is responsible for matching the name of the IO. DS - for matching the name of the set of IO corresponding to the name. FS is responsible for the transport of IO. SS is responsible for placing the IO in the network.

In the proposed approach to organizing NS, NST is part of NS, which becomes possible when special requirements for NS are presented. In addition, the choice of the NS construction method determines the solution to the previously described problem of reducing the convergence rate when scaling a distributed system.

In the case of prefix routing, there is a relationship between the number of routing rules and the number of IOs. An example of a routing approach that solves the problem of such a connection is label routing, in which the number of labels depends on the topology of the network and does not depend on the number of IOs in the network. In this approach, a label corresponds to each node, and the route is represented by a sequence of labels. However, in the case of labels, the labels themselves are not connected in any way with the IO, so there is a need for a mechanism that determines the location of the IO, and the problem is transferred to the level of this mechanism.

Thus, it is necessary that the labels be associated with the IO, for example, through the names of the IO, but at the same time, so that the number of labels does not depend on the number of IOs. It is possible to observe these conditions if the names of the IO will form a space that can be divided into areas, each of which will be associated with a label. In this case, the label will correspond to the area of the name space, and the routing rule will correspond to the transition between the areas. To ensure that the number of routing rules does not depend on the number of IOs, it is necessary that the number of areas does not change when the number of IOs changes when the system is scaled.

If one fix the number of areas, this will mean that the name space is finite. Thus, it is necessary to embed a potentially infinite number of IOs in a finite name space. The aim of the proposed approach is to compare the IO of the “place” in such a space should occur automatically. In addition, you must be able to "navigate" in such a space in order to be able to access the IO, not knowing exactly where it is in this space.

Since space is a logical object, it is necessary to map the areas that make up this space onto a distributed system. Connectivity between neighboring areas in space is accomplished using routing rules between nodes. Thus, one is talking about an overlay network.

This work does not discuss the problem of dividing space into areas and constructing a mapping of areas onto a physical topology — the task of organizing an MST. Such a task can be posed as an optimization problem, where it is necessary to minimize the number of “physical” routing rules in DST, provided that the connectivity between neighboring regions in the logical space defined in NST is respected. However, the solution to the problem is not relevant for any name space, but only for the finite, because otherwise, when scaling a distributed system, the quality of such a mapping will gradually deteriorate.

3.1 Name space organization

The key idea for organizing the name space is related to the work [4], which proposed the concept of perceptual hashes, such that visually similar objects map close hashes. Clearly, one is talking about compressing an image of an arbitrary size in an image of a fixed size, which is a hash. In fact, such a representation is used that is invariant to a number of transformations. In relation to ICN, in the case of work with text objects, this means that the meaningful IOs have similar names. This key property for organizing NS is called the property of semantic similarity.

The observance of the semantic similarity property allows one to organize effective access to IO, since it does not require additional costs: knowing the distance from the neighbors of a certain IO to the desired IO, one must "move" towards decreasing the distance. The “navigation” process can be called semantic routing, and the way of choosing a direction is called semantic gradient. Considering the above, the property of semantic similarity was chosen as a requirement for NS. Close analogues of semantic routing are hyperbolic routing [5] and kademlia routing scheme[6].

In order to formalize the requirement of semantic similarity, the statement of the Distance Geometry Problem (DGP) problem [7] was originally used, in which the distance matrix between the elements of the set required the coordinates of points in the Euclidean space of the smallest dimension, such that the distance matrix between the points exactly coincides with original distance matrix. With regard to ICN, this means that distances express semantic similarity between pairs of IO, and the points correspond to the names of IO.

Unfortunately, such a formulation makes it possible to find out only whether a solution exists, but does not allow one to construct it, because despite the existence of a computationally efficient procedure for obtaining the coordinates of points, this procedure can be applied only if the desired mapping exists for the original matrix. In other words, it is necessary that the original similarity matrix be Euclidean. Despite the existence of a computationally efficient procedure for checking the original similarity matrix for Euclideanity, in most cases the verification fails, since semantic similarity may not satisfy the triangle inequality. For example, if the similarity of the object A to the object B is 7, the object B to the object C 4, and C to A — 2, then the sides of the triangle cannot make such values. The Euclidean requirement is redundant, it is enough that the target space is metric, however, it is for Euclidean spaces that the methods for solving the problem are most studied.

The statement of the problem, which allows one to check not only the existence of the solution, but also to construct it, is connected with the multidimensional scaling problem (MDS) [8], where it is also necessary to find the coordinates of points in the Euclidean space using the given similarity matrix, however, besides the statement with the exact solution, cMDS , there is also a statement with an approximate solution, mMDS, i.e. the task is posed as optimization.

Moreover, in multidimensional scaling there is also an nMDS task in which it is required to preserve not their original differences, but their ranks, i.e. those objects that are "further" must be displayed at farther points, and those that are "closer" — at closer ones. In other words, it is necessary to observe monotony. This task is also an optimization problem. An example of the nMDS problem is in the competition [9]. This is a key task for building a name space.

3.2 Build word representations

Since the number of texts is potentially infinite, the nMDS problem is solved for words, since one can assume that the words are finite, which means that the word similarity matrix will not increase when the number of IO changes. The dictionary of words is determined by the selected corpus of texts. For example, the Taiga text corpus contains 5 billion words, of which about 250,000 are unique [10]. The size of the full similarity matrix in this case is about 500Gb for float64. The word similarity matrix can be obtained from the corpus of texts or calculated using pre-calculated representations of words, for example, using word2vec [11].

Approaches to solving the nMDS problem are not the subject of this article. In this problem, given a complete matrix of similarity or a given function that calculates similarity for any two IOs, it is necessary to construct a map in a Euclidean space of fixed dimension, preserving the ranks of distances as accurately as possible. It is important to note that it is possible to use the Spearman correlation coefficient (Scc) not only as an indicator of the quality of the solution, but also as a function of losses.

The complexity of using Scc as a loss function is due to the fact that the argsort function used to calculate it, which determines the similarity indices in an ordered sequence of similarity, has no derivative. Another difficulty is related to the fact that the calculation of Scc for a full similarity matrix is computationally inefficient, therefore, it is required to propose a method for approximating Scc.

The result of solving the nMDS problem for words is to obtain representations of words in the word space such that the orders of distances in the matrix of distances between points correspond as closely as possible to orders of similarity in the original similarity matrix. The quality of conformity of the orders is expressed by Scc.

3.3 Using ION

If one takes 3 as the target dimension of space, then the resulting representations can be visualized. In this case, the word will correspond to a point in three-dimensional space. In addition, if it is enough to accurately preserve the ranks of similarity, it may turn out that the coordinate components will be interpreted, since the words will be divided into semantic clusters.

If one consider text objects as an IO, they can be represented in a three-dimensional word space in the form of trajectories corresponding to the sequences of words that make up the texts. In this case, such trajectories can be perceived as ION, since users can work with them. In particular, the user may have the task of finding the most similar texts for the given text.

3.4 NID application

A nonconstructive statement of the definition problem for a given trajectory in the word space of the nearest trajectories is to construct a mapping of the smallest length between all pairs of trajectories, word mover's distance (WMD) [12]. Despite the existence of a computationally efficient procedure for calculating the minimum length of a mapping for a pair of trajectories, it is not possible to determine the minimum on the entire set of trajectories, since in this case a complete enumeration of all trajectories will be required, the number of which can increase potentially infinitely.

A constructive statement of the problem of determining trajectories is the use of the Tube method, based on the use of the inverse index. The idea of the method is that it is possible to lay a "pipe" around the selected path and those paths that entirely enter the tunnel and will be candidates for the nearest path. If for each word a list of words is ordered by distance from a given word, then it is possible to specify a "radius" of deletion, for example, the first 100 words from the beginning of the list. Then, if for each word also store the inverse index of the texts in which the given word is included, then the search for the trajectories closest to the given one is reduced to the intersection of the union of the inverse indices of the words of the given trajectory. Combining and intersecting inverse indices can be performed computationally efficiently.

It is important to note that the reverse index of words does not store the text objects themselves, but their identifiers, NID. In addition, NIDs can be set not arbitrarily, but so that close text paths have similar NIDs. This can increase the efficiency of the operation of crossing the NID lists, because if the texts are grouped by meaning, the NIDs form "condensations" on the number line and it will be possible not to see the "voids" between them when crossing the NID lists.

4 EVALUATION

The questions of further research are the relationship of the dimension of representations with the architecture of the system, the ability to use embeddings to represent computations, generalization of the proposed technique to other types of IO, in particular images, and the question of how to calculate the distance between sets of trajectories.

5 CONCLUSION

The paper proposes an approach to naming IO, in which objects similar in meaning are located nearby in the coordinate space. Perhaps, using the embeddings, it is possible to organize the New Internet, since the space of meanings is uniform and does not depend on the form of representation of the IO.

ACKNOWLEDGMENTS

A.I. Maysuradze for setting the isotonic embedding problem.

REFERENCES

[1] Van Jacobson, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs, and Rebecca L. Braynard. 2009. Networking named content. In Proceedings of the 5th international conference on Emerging networking experiments and technologies (CoNEXT '09). Association for Computing Machinery, New York, NY, USA, 1–12.

[2] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K.H. Kim, S. Shenker, I. Stoica, A data-oriented (and beyond) network architecture, in: Proceedings of ACM SIGCOMM'07, Kyoto, Japan, August 2007.

[3] Lixia Zhang, Alexander Afanasyev, Jeffrey Burke, Van Jacobson, kc claffy, Patrick Crowley, Christos Papadopoulos, Lan Wang, and Beichuan Zhang. 2014. Named data networking. SIGCOMM Comput. Commun. Rev. 44, 3 (July 2014), 66–73.

[4] Zauner, Christoph: Implementation and Benchmarking of Perceptual Image Hash Functions. Master's thesis, Upper Austria University of Applied Sciences, Hagenberg Campus, 2010.

[5] V. Lehman et al., "An experimental investigation of hyperbolic routing with a smart forwarding plane in NDN," 2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS), Beijing, 2016, pp. 1-10.

[6] Maymounkov, Petar and Eres, David. (2002). Kademlia: A Peer-to-peer Information System Based on the XOR Metric.

[7] Blumenthal, L.M. (1970). Theory and applications of distance geometry (2nd ed.). Bronx, New York: Chelsea Publishing Company. pp. 90–161.

[8] Szubert, B., Cole, J.E., Monaco, C. et al. Structure-preserving visualisation of high dimensional single-cell datasets. Sci Rep 9, 8914 (2019).

[9] Kaggle. Implementation distances ranks.
https://www.kaggle.com/c/nicn2 (accessed June 29, 2020).

[10] Kutuzov A., Kuzmenko E. (2017) WebVectors: A Toolkit for Building Web Interfaces for Vector Semantic Models. In: Ignatov D. et al. (eds) Analysis of Images, Social Networks and Texts. AIST 2016. Communications in Computer and Information Science, vol 661. Springer, Cham

[11] Mikolov, Tomas and Corrado, G.s and Chen, Kai and Dean, Jeffrey. (2013). Efficient Estimation of Word Representations in Vector Space. 1-12.

[12] M. J. Kusner, Y. Sun, N. I. Kolkin, and K. Q. Weinberger. From word embeddings to document distances. In ICML, 2015.