Results of the Sequence PredIction ChallengE (SPiCe): a Competition on Learning the Next Symbol in a Sequence

Borja Balle B.DEBALLEPIGEM@LANCASTER.AC.UK
Lancaster University, United Kingdom

Rémi Eyraud REMI.EYRAUD@LIF.UNIV-MRS.FR
QARMA team, LIF, France

Franco M. Luque FRANCOLQ@FAMAF.UNC.EDU.AR
Universidad Nacional de Córdoba and CONICET, Argentina

Ariadna Quattoni ARIADNA.QUATTONI@XRCE.XEROX.COM
Xerox Research Center Europe

Sicco Verwer S.E.VERWER@TUDELFT.NL
Delft University of Technology, The Netherlands

Abstract

The Sequence PredIction ChallengE (SPiCe) is an on-line competition that took place between March and July 2016. Each of the 15 problems was made of a set of whole sequences as training sample, a validation set of prefixes, and a test set of prefixes. The aim was to submit a ranking of the 5 most probable symbols to be the next symbol of each prefix.

1. Introduction

The context of the Sequence PredIction ChallengE (SPiCe) is the one of learning from symbolic sequences, that is from strings of symbols. This competition is part of a long list of such challenges, among which we can cite ABBADINGO (1998) on learning deterministic finite state automata [7], OMPHALOS (2004) on learning context-free grammars [3], TENJINNO (2006) on learning finite state transducers [10], ZULU on active learning of finite state machines [4], and PAUTOMAC on learning probabilistic finite state machines [12].

The goal of SPiCe is to gather together researchers from various fields around a common problem: guessing the next symbol in a sequence. This issue arises in applications of many domains, from natural language processing to bioinformatics, including software engineering and many others.

2. SPiCe data

The competition is made of 15 problems: 4 purely synthetic, 4 partly synthetic, and 7 real-world data. Table 1 summarizes the characteristics of the different problems.

In more detail, Problems 1, 2, and 3 were artificially generated following the same approach: we constructed an HMM with n states and non-stationary transition probabilities by partitioning the unit interval $[0, 1)$ into n equal sub-intervals and letting the states evolve

Number	Alphabet	Train	Test	Type
1	20	20000	5000	synthetic (non-stationary HMM with 2 states)
2	10	20000	5000	synthetic (non-stationary HMM with 2 states)
3	10	20000	5000	synthetic (non-stationary HMM with 4 states)
4	33	5987	749	NLP (English verbs, character level, Penn Treebank)
5	49	33654	4207	NLP (character level language modeling, Penn Treebank)
6	60*	5000	5000	partly synthetic, software engineering (RERS 2013 problem 34)
7	20	65438	5000	biology (protein family PF13855, full set, Pfam)
8	48	13903	1738	NLP (Spanish simplified POS sentences, Ancora)
9	11*	5000	5000	partly synthetic, software engineering (RERS 2013 problem 42)
10	20	54932	4848	biology (protein family PF00400, RP15 subset, Pfam)
11	6722	32384	4048	NLP (English lemmas from Flickr-8000)
12	21	200000	3000	synthetic (PAutomaC generator)
13	702	26544	3318	NLP (English spelling correction from Twitter Typos Corpus)
14	27	10000	5000	partly synthetic (ALERGIA, DFA based on problem 4)
15	32	50000	5000	partly synthetic (ALERGIA, DFA based on problem 5)

Table 1: SPiCe data characteristics. The column *Alphabet* gives the number of different symbols (the star indicates a problem for which this value is only an upper bound), *Train* and *Test* provides respectively the number of elements in the training and test sets, *Type* details the source of the data.

as $h_{t+1} = h_t + \phi \mod 1$, for some irrational number ϕ. The emission probabilities were sampled from a Dirichlet distribution.

Problems 4, 5, and 8 all correspond to NLP data from Penn Treebank [9] and the Spanish Ancora corpus [11]. Problem 11 was created from a lemmatized version of the Fickr-8k dataset [8]. Problem 13 was derived from a Twitter spelling correction corpus [1].

Problems 6 and 9 are synthetic data based on software engineering ones and come from the challenge RERS 2013 [6].

Problems 7 and 10 are protein families sequences taken from the Pfam database [5].

Problem 12 consists of synthetic data generated using the PAutomaC data generator [12].

Finally, Problems 14 and 15 contain synthetic data generated from two Deterministic Finite State Automata learned using the ALERGIA algorithm [2] on the NLP data sets of Problems 4 and 5, respectively.

For each problem, SPiCe provides 3 data sets:

- A training sample that contains whole sequences,
- A validation sample, called *public test set*, that contains prefixes of sequences,
- A test sample, called *private test set*, that contains prefixes of sequences.

From the training sample, a participant can learn a model. Then, (s)he can submit rankings for all prefixes of the public test set. When a ranking has been send for all prefixes, (s)he receives as a feedback the score of the submission (see Section 3 for details about the score computation) and is allowed to submit again on the test set. Once the participant decided the tuning phase is over, (s)he can submit on the private test set. Only one submission is allowed on that test set and no submission can be made on that problem once this submission is accomplished.

In order to ensure that the participants are not using the private test set while learning their model, which would create an important bias, only the first prefix of the set is available beforehand. When a participant submits its ranking for this prefix, he is fed with the next prefix for which he has to submit a ranking. The process continues until all prefixes have been sent to the participant. For obvious reasons of consistency, the same process is used for the public test set.

3. Evaluation Metrics

The SPiCe competition focuses on the ability of the learned models to predict the next symbol in a string.

For any given task, the participants are given a test set $T = (y_1, \dots, y_M)$ and for each string y_i they are asked to produce a ranking (i.e. an ordered list) of 5 possible next symbols[1] $\hat{n}(y_i) = (\hat{a}^i_1, \dots, \hat{a}^i_5)$ sorted from more likely to less likely.

The predictions $\hat{n}(y_i)$ is evaluated using a ranking metric based on *normalized discounted cumulative gain at* 5. To compute this metric we assume that given a prefix y_i we know a probability distribution $p(\bullet|y_i)$ for the next symbol. Then the metric is given by

$$NDCG_5(\hat{a}^i_1, \dots, \hat{a}^i_5) = \frac{\sum_{k=1}^{5} \frac{p(\hat{a}^i_k|y_i)}{\log_2(k+1)}}{\sum_{k=1}^{5} \frac{p_k}{\log_2(k+1)}} \ , \qquad (1)$$

where $p_1 \geq p_2 \geq \cdots \geq p_5$ are the top 5 values in the distribution $p(\bullet|y_i)$. Note this makes sure the score is normalized between 0 and 1. The closer to 1 the evaluation is, the better (in the worst case it is valued to 0).

The distributions $p(\bullet|y)$ is computed differently depending on whether the data is synthetic (and the model that generated it is available) or real.

For synthetic data we use the true conditional distribution over the next symbol.

For real data, where the string y is obtained as a prefix of a longer string yax, we take $p(a'|y) = \delta_{a=a'}$. Note that in this case we have $p_1 = 1$ and $p_2 = \cdots = p_5 = 0$. Thus, when applying this metric to real data we get $NDCG_5(\hat{a}_1, \dots, \hat{a}_K) = 1/\log_2(\hat{k}+1)$, where $\hat{k}$ is such that $\hat{a}_{\hat{k}} = a$ (and $\hat{k} = \infty$ if a is not in the list of predicted next symbols). Note that if the prefix y_i appears multiple times in the test set, then the participants are asked to predict a next symbol for it multiple times. The true next symbol in each of these real sequences might be different every time, thus giving different evaluations of the ranking produced every time, and in the limit this gives the expected result of comparing the ranking with the expected distribution over the next symbol.

The score of a submission is the sum of the $NDCG_5$ on each prefix divided by the number of prefixes in the test set.

[1] The possible next symbols are the ones seen in the training set and a special one marking the end of the word (its probability is thus the probability of the prefix y_i to be a whole sequence)

4. Baselines.

We provide 2 baselines and script examples to submit rankings: a toolbox in spectral learning of weighted automata and a 3-gram.

5. Results

Competition Activity. 82 participants registered to have access to the data and 26 of them submitted at least one of their solutions to a problem. There were a total number of 3698 complete submissions on a public test set and 9 teams submitted on all private test sets (a 10th team submitted to all but one of these test sets).

Overall results. The final results of the competition are given in Table 2 in Annex. The detailed scores on each problem are available on the website. There is a clear winner of SPiCe: team *shib* of Chihiro Shibata. They finished first on all NLP and software engineering data, second on the 2 biology data, and in the top 2 on all synthetic data. Surprisingly, their approach is among the less efficient on the synthetic from NLP data. The team that finished second is *ToBeWhatYouWhatToBe* of Shanbo Chu. Their approach enjoyed impressive stable results: they were in the top 5 of all problem but two (6th on both the synthetic Problem 12 and on the software engineering Problem 9). Team *ushitora* of Ichinari Sato, who completed the podium, is the only one that defeated regularly the winning team: they did better on 5 problems. Their relatively bad results on two NLP data sets and on the synthetic from NLP ones costed them the victory.

We notice that the results are tight: there is about a point between the score of the first team and the one of the fourth (that corresponds to less than 7% variation) and the variation is 10% between the first and the eighth team.

6. Conclusion

The results of SPiCe presented in this paper indicate that the competition was fruitful: refined methods for learning the next symbol in a sequence have been designed and a detailed comparison of their performances is available.

The disclosure of the content of each method will be a very interesting moment and will certainly yield to a deeper understanding of string distribution learning algorithms.

References

[1] Twitter typo corpus, url: http://luululu.com/tweet/.

[2] Rafael C. Carrasco and Jose Oncina. Learning stochastic regular grammars by means of a state merging method. In *Proc. 2nd International Colloquium in Grammatical Inference, ICGI*, pages 139–152, 1994.

[3] Alexander Clark. Learning deterministic context free grammars: The Omphalos competition. *Machine Learning*, 66(1):93–110, 2007.

[4] David Combe, Colin de la Higuera, and Jean-Christophe Janodet. Zulu: An interactive learning competition. In *Proc. 8th International Conference on Finite-state Methods and Natural Language Processing*, FSMNLP'10, pages 139–146, 2010.

[5] Robert D. Finn, Penelope Coggill, Ruth Y. Eberhardt, Sean R. Eddy, Jaina Mistry, Alex L. Mitchell, Simon C. Potter, Marco Punta, Matloob Qureshi, Amaia Sangrador-Vegas, Gustavo A. Salazar, John Tate, and Alex Bateman. The pfam protein families database: towards a more sustainable future. *Nucleic Acids Research*, 44(D1):D279–D285, January 2016.

[6] Falk Howar, Malte Isberner, Maik Merten, Bernhard Steffen, Dirk Beyer, and CorinaS Păsăreanu. Rigorous examination of reactive systems. 16(5):457–464, 2014. doi: 10.1007/s10009-014-0337-y.

[7] Kevin Lang, Barak A. Pearlmutter, and Rodney Price. Results of the abbadingo one dfa learning competition and a new evidence driven state merging algorithm. In *Proc. 4th International Colloquium on Grammatical Inference, ICGI*, 1998.

[8] P. Young M. Hodosh and J. Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013.

[9] Mitchell P. Marcus, Beatrice Santorini, and Mary A. Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2): 313–330, 1994.

[10] Bradford Starkie, Menno van Zaanen, and Dominique Estival. The tenjinno machine translation competition. In Yasubumi Sakakibara, Satoshi Kobayashi, Kengo Sato, Tetsuro Nishino, and Etsuji Tomita, editors, *Proc. 8th International Colloquium on Grammatical Inference, ICGI*, pages 214–226, 2006.

[11] Mariona Taulé, Maria A. Martí, and Marta Recasens. AnCora: Multilevel annotated corpora for catalan and spanish. In *Proceedings of the International Conference on Language Resources and Evaluation*, Marrakech, Morocco, 2008.

[12] Sicco Verwer, Rémi Eyraud, and Colin de la Higuera. Pautomac: a probabilistic automata and hidden markov models learning competition. *Machine Learning*, 96 (1-2):129–154, 2014. ISSN 0885-6125. doi: 10.1007/s10994-013-5409-9.

Annex

Position	Team Name	Head of the Team	Global Score
1	shib	Chihiro Shibata	10.4498481750
2	ToBeWhatYouWhatToBe	Shanbo Chu	10.0198711157
3	ushitora	Ichinari Sato	9.9562549591
4	Markov_s_Principle	Farhana Ferdousi Liza	9.4082437158
5	ZZZZZZZZ	Du Xi	9.1841279864
6	uwtacoma	Martine De Cock	9.0242664516
7	Ping	Benjamin Loos	8.9515981674
8	vha	Quang Vinh Dang	8.9494856894
9	Rafael-UoL	Rafael Ktistakis	6.7270460427
10	TeamEigen	Alok Kumar	6.0221453160

Table 2: SPiCe final results of the top 10 participants. The global score is the sum of the scores on the 15 SPiCe problems computed on the rankings given on the private test sets.