A Machine Learning Approach to Acronym Generation

Yoshimasa Tsuruoka[†‡]
[†]CREST
Japan Science and Technology Agency
Japan

Sophia Ananiadou
School of Computing
Salford University
United Kingdom

Jun'ichi Tsujii[‡†]
[‡]Department of Computer Science
The University of Tokyo
Japan

tsuruoka@is.s.u-tokyo.ac.jp
S.Ananiadou@salford.ac.uk
tsujii@is.s.u-tokyo.ac.jp

Abstract

This paper presents a machine learning approach to acronym generation. We formalize the generation process as a sequence labeling problem on the letters in the definition (expanded form) so that a variety of Markov modeling approaches can be applied to this task. To construct the data for training and testing, we extracted acronym-definition pairs from MEDLINE abstracts and manually annotated each pair with positional information about the letters in the acronym. We have built an MEMM-based tagger using this trainig data and evaluated the performance of acronym generation. Experimental results show that our machine learning method gives significantly better performance than that achieved by the popular heuristic rule for acronym generation and enables us to obtain multiple candidate acronyms together with their likelihoods represented in probability values.

1 Introduction

One of the simplest way to generate acronyms from definitions is to choose the letters at the beginning of each word and capitalize them. However, there are a lot of exceptions in the acronyms appearing in biomedical documents. The followings are some real examples of the definition-acronym pairs that cannot be created with the simple heuristic method.

RNA polymerase (RNAP)
bioconcentration factor (BF)
melanoma cell adhesion molecule (Mel-CAM)
the xenoestrogen 4-tert-octylphenol (t-OP)

In this paper we present a machine learning approach to automatic generation of acronyms from the given expanded forms. We formalize this problem as a sequence labeling task such as part-of-speech tagging, chunking and other natural language tagging tasks so that a common Markov modeling approache can be applied to this task.

2 Acronym Generation as a Sequence Labeling Problem

Given the definition (expanded form), the mechanism of acronym generation can be regarded as the task of selecting the appropriate action on each letter in the definition.

Figure 1 illustrates an example, where the definition is "Duck interferon gamma" and the generated acronym is "DuIFN-gamma". The generation proceeds as follows:

> The acronym generator outputs the first two letters unchanged and skips the following three letters. Then the generator capitalizes 'i' and skip the following four letters...

By assuming that an acronym is made up of alphanumeric letters, spaces and hyphens, the actions being taken by the generator are classified into the following five classes.

Definition	Actions
D	UPPER
u	LOWER
c	SKIP
k	SKIP
	SKIP
i	UPPER
n	SKIP
t	SKIP
e	SKIP
r	SKIP
f	UPPER
e	SKIP
r	SKIP
o	SKIP
n	UPPER
	HYPHEN
g	LOWER
a	LOWER
m	LOWER
m	LOWER
a	LOWER

Acronym
D
u
I
F
N
-
g
a
m
m
a

Figure 1: Acronym generation as a sequence labeling problem. The definition is “Duck interferon gamma” and the acronym is “DuIFN-gamma”. Each letter in the acronym is generated from a letter in the definition following the action for the letter.

- SKIP

  The generator skips the letter.

- UPPER

  If the target letter is uppercase, the generator outputs the same letter. If the target letter is lowercase, the generator coverts the letter into the corresponding upper letter.

- LOWER

  If the target letter is lowercase, the generator outputs the same letter. If the target letter is uppercase, the generator coverts the letter into the corresponding lowercase letter.

- SPACE

  The generator convert the letter into a space.

- HYPHEN

  The generator convert the letter into a hyphen.

From the probabilistic modeling point of view, this task is to find the sequence of actions $t_1...t_n$ that maximizes the following probability given the observation $o = o_1...o_n$

$$P(t_1...t_n|o). \tag{1}$$

Observations are the letters in the definition and various types of features derived from them. We decompose the probability in a left-to-right manner.

$$P(t_1...t_n|o) = \prod_{i=1}^{n} p(t_i|t_1...t_{i-1}o). \tag{2}$$

By making a first-order markov assumption, the equation becomes

$$P(t_1...t_n|o) = \prod_{i=1}^{n} p(t_i|t_{i-1}o). \tag{3}$$

If we have the trainig data containing a large number of definition-acronym pairs where the definition is annotated with the labels for actions, we can estimate the parameters of this probabilstic model and the best action sequence can be efficiently computed by using a Viterbi decoding algorithm.

In this paper we adopt a maximum entropy model (Berger et al., 1996) to estimate the local probabilities $p(t_i|t_{i-1}o)$ since it can incorporate diverse types of features with reasonable computational cost. This modeling, as a whole, is called Maximum Entropy Markov Modeling (MEMM).

Regularization is important in maximum entropy modeling to avoid overfitting to the training data. For this purpose, we use the maximum entropy modeling with inequality constraints (Kazama and Tsujii, 2003). The model gives equally good performance as the maximum entropy modeling with Gaussian priors (Chen and Rosenfeld, 1999), and the size of the resulting model is much smaller than that of Gaussian priors because most of the parameters become zero. This characteristic enables us to easily handle the model data and carry out quick decoding, which is convenient when we repetitively perform experiments. This modeling has one parameter to tune, which is called *width factor*. We set this parameter to be 1.0 throughout the experiments.

3 The Data for Training and Testing

Since there is no training data available for the machine learning task described in the previous section, we manually created the data. First, we extracted definition-acronym pairs from MEDLINE abstracts using the acronym acquisition method proposed by (Schwartz and Hearst, 2003). The abstracts used for constructing the data were randomly selected from the abstracts published in the year of 2001. Duplicated pairs were removed from the set.

In acquiring the pairs from the documents, we focused only on the pairs that appear in the form of

... *expanded_form* (*acronym*) ...

We then manually removed misrecognized pairs, and annotated each pair with positional information. The positional information tells which letter in the definition should correspond to a letter in the acronym. Table 1 lists a portion of the data. For example, the positional information in the first pair indicates that the first letter 'i' in the definition correspods to 'I' in the acronym, and the 12th letter 'm' corresponds to 'M'.

With this positinal information, we can create the training data for the sequence labeling task because there is one-to-one correspondence between the sequence labels and the data with positional information. In other words, we can determine the appropriate label for each letter in the definition by comparing the letter with the corresponding letter in the acronym.

Definition	Acronym	Positional Information
intestinal metaplasia	IM	1, 12
lactate dehydrogenase	LDH	1, 9, 11
cytokeratin	CK	1, 5
cytokeratins	CKs	1, 5, 12
Epstein-Barr virus	EBV	1, 9, 14
30-base pairs	bp	4, 9
in-situ hybridization	ISH	1, 4, 9
:	:	:

Table 1: Curated data containing defintions, thier acronyms and the positional information.

4 Features

Maximum entropy modeling allows us to incorporate diverse types of features. In this paper we use the following types of features in local classification. As an example, consider the situation where we are going to determine the action at the letter 'f' in the definition "Duck interferon gamma".

- Letter unigram

  The unigrams of neighboring letters. (e.g. 'r', 'f', 'e')

- Letter bigram

  The bigrams of neighboring letters. (e.g. "er", "rf", "fe", "er")

- Letter trigram

  The trigrams of neighboring letters. (e.g. "ter", "erf", "rfe", "fer", "ero")

- Letter sequence

  1. The sequence of letters ranging from the beginning of the word to the target letter. (e.g. "interf")
  2. The sequence of letters ranging from the target letter to the end of the word. (e.g. "feron")

Rank	Probability	String
1	**0.779**	**TBI**
2	0.062	TUBI
3	0.028	TB
4	0.019	TbI
5	0.015	TB-I
6	0.009	tBI
7	0.008	TI
8	0.007	TBi
9	0.002	TUB
10	0.002	TUbI
ANSWER		TBI

Table 2: Generated acronyms for "traumatic brain injury".

Rank	Probability	String
1	0.423	ORF1
2	0.096	OR1
3	**0.085**	**ORF-1**
4	0.070	RF1
5	0.047	OrF1
6	0.036	OF1
7	0.025	ORf1
8	0.019	OR-1
9	0.016	R1
10	0.014	RF-1
ANSWER		ORF-1

Table 3: Generated acronyms for "open reading frame 1".

- Distance
  1. The distance between the target letter and the beginning of the word. (e.g. 6)
  2. The distance between the target letter and the tail of the word. (e.g. 5)
- Definition Length

  The number of words in the definition (e.g. 3)
- Action history

  The preceding action (e.g. SKIP)

5 Experiments

To evaluate the performance of the acronym generation method presented in the previous section, we ran five-fold cross validation experiments using the manually curated data set. The data set consists of 1,901 definition-acronym pairs.

For comparsion, we also tested the performance of the popular heuristics for acronym generation in which we choose the letters at the beginnings of each word in the definition and capitalize them.

5.1 Features

To evalute how much individual types of features affect the generation performance, we ran experiments using different feature templates. Table 7 shows the results. Overall, the results show that various types of features have been successfully incorporated in the MEMM modeling, leading to improved performance.

Rank	Probability	String
1	0.163	RNA-P
2	0.147	RP
3	0.118	RNP
4	**0.110**	**RNAP**
5	0.064	RA-P
6	0.051	R-P
7	0.043	RAP
8	0.041	RN-P
9	0.034	RNA-PM
10	0.030	RPM
ANSWER		RNAP

Table 4: Generated acronyms for "RNA polymerase".

The performance achieved with only unigram features is almost the same as that achieved by the heuristic rule. Note that the features on the preivous state improve the performance, which suggests that our selection of the states in the Markov modeling is a reasonable choice for this task.

5.2 Influential Features

In the maximum entropy modeling, you can grasp influential features by examining the weights of features[1]. Table ? shows some features that gained a large weight as a result of training.

[1]Care has to be taken when you look at the weights of features because overlapping of features affects the weights. For example, if you defi ne two identical features, the weights of the individual features are halved.

Feature Templates	Top 1 Coverage (%)	Top 5 Coverage (%)	Top 10 Coverage (%)
UNI	48.2	66.2	74.2
UNI, BI	50.1	71.2	78.3
UNI, BI, TRI	50.4	72.3	80.1
UNI, BI, TRI, HIS	50.6	73.6	81.2
UNI, BI, TRI, HIS, ATH	51.0	73.9	80.9
UNI, BI, TRI, HIS, ATH, LEN	53.9	74.6	81.3
UNI, BI, TRI, HIS, ATH, LEN, DIS	54.4	75.0	81.8
UNI, BI, TRI, HIS, ATH, LEN, DIS, SEQ	55.1	75.4	82.2

Table 7: Performance with Different Feature Sets.

Rank	Probability	String
1	0.405	MCPP
2	0.149	MCP
3	0.056	MCP
4	0.031	MPP
5	0.028	McPP
6	0.024	MchPP
7	0.020	MC
8	0.011	MP
9	**0.011**	**mCPP**
10	0.010	MCRPP
ANSWER		mCPP

Table 5: Generated acronyms for "meta-chlorophenylpiperazine".

Rank	Coverage (%)
1	55.2
2	65.8
3	70.4
4	73.2
5	75.4
6	76.7
7	78.3
8	79.8
9	81.1
10	82.2
BASELINE	47.3

Table 6: Coverage achieved with the Top N Candidates.

It is interesting that the accuracy achieved by the heuristic rule is ??.

5.3 Learning Curve

5.4 Error Analysis

6 Discussion

7 Conclusion

We presetned a machine learning approach to acronym generation. In this approach, we regarded the generation process as a sequence labeling problem like POS tagging, and we manually created the data for training and testing.

Experimental results using 1901 pairs, we achieved a coverage (also accuracy) of 55.1%, which is significantly bettern than that achieved by the popular heuristics for acronym generation. The

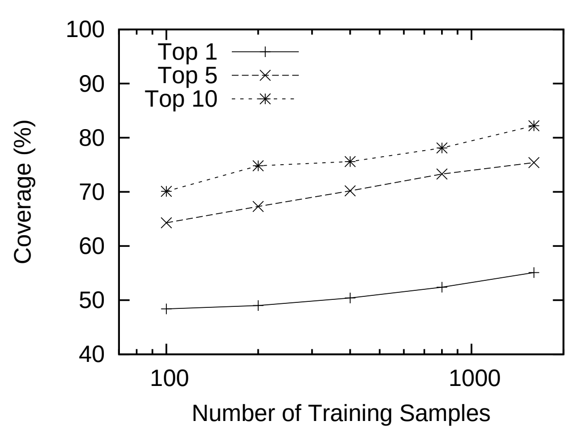

Figure 2: Learning curve.

algorithm also enables us to have other acronym candidates together with the probabilities representing their likelihood.

In this paper we did not consider the generation patterns where the letters in the acronym appear in a different order in the definition (e.g. ??? for ???). Since about ??% of acronyms involve this types of generation mechanism, we might further improve performance by considering such permutation of letters.

The leraning curve (Fig 2) suggests that we will have improved performance if we have more training data. The size of the training data used in the experiments is fairly small compared to those in other sequence tagging tasks such POS tagging and chunking. We plan to increase the size of the training data with a semi-automatic way that could reduce the human effort for annotation.

References

Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Stanley F. Chen and Ronald Rosenfeld. 1999. A gaussian prior for smoothing maximum entropy models. *Technical Report CMUCS -99-108, Carnegie Mellon University*.

Jun'ichi Kazama and Jun'ichi Tsujii. 2003. Evaluation and extension of maximum entropy models with inequality constraints. In *Proceedings of EMNLP 2003*.

Ariel Schwartz and Marti Hearst. 2003. A simple algorithm for identifying abbreviation definitions in biomedical texts,. In *Proceedings of the Pacific Symposium on Biocomputing (PSB 2003)*.