Context based number normalization using skip-chain conditional random fields

Linas Balčiūnas
Vytautas Magnus University
Kaunas University of Technology
Kaunas, Lithuania
linasb20@gmail.com

***Abstract*—Verbalizing numeric text tokens is a required task for various speech-related applications, including automatic speech recognition and text-to-speech synthesis. In morphologically rich languages, such conversion involves predicting implicit morphological properties of a corresponding numeral. In this paper, we propose first-order skip-chain Conditional Random Field (CRF) models and various prepossessing techniques to leverage different contextual information. We show that our best skip-chain CRF models achieve over 80% accuracy on the set of 2000 Lithuanian sentences.**

***Keywords*—Number normalization, text normalization, conditional random field, natural language processing**

I. Introduction

Number normalization is the task of replacing numeric tokens in a sentence by numerals (word tokens) using an appropriate inflected form of a numeral. Number normalization usually involves disambiguation as the same numeric token needs to be mapped into different word forms depending on the context (e.g. '5 vaikai eina' ↦ '*Penki* vaikai eina' (five children are going) vs. '5 vaikų nėra' ↦ '*Penkių* vaikų nėra' (five children are missing)). Although number normalization can be considered as a part of a broader task of text normalization, formulating it as the separate task might be beneficial, since the process of number normalization can be quite complex depending on morphological features of a language. In this paper, we describe the process of building and evaluating a number normalization system for Lithuanian. However, some techniques and models are language-independent and might be applied to other languages. In Lithuanian, for example, number '5' depending on sentence context may represent any of 63 different words. Predicting this relationship directly is rather difficult and it would require huge data-set to properly learn Numeral grammar. A simpler approach is predicting Part Of Speech (POS) tag and then generating numerals accordingly. POS tag contains all necessary morphological information to use language-specific grammar based numbers-to-words system [1]. This way possible result classes are shared across all numbers and predicting POS tag can be formulated as sequence labeling, rather than sequence-to-sequence task, because of the one-to-one relationship.

II. Related works

As far as we know, there are no published works or publicly available applications performing number normalization based on the sentence context for Lithuanian. Although, many languages deal with similar morphological disambiguation problems. Russian and Lithuanian numbers share many morphological properties, including types (cardinal, ordinal), genders and cases. There are existing research and systems for the Russian language on general text normalization, handwritten language-general grammar [2], [3], Recurrent Neural Network (RNN) [4], and number normalization [5], [6], [7].

III. Preprocessing

A. Data

A small text corpus for training and evaluating text normalization models was collected and manually annotated. It consists of 1955 sentences containing 3143 numeric tokens. Sentences were inspected by linguists who suggested a numeral word form as an appropriate replacement for every numeric token. In some ambiguous cases, a few reasonable alternatives were proposed. Some ambiguities were related to the use of the pronominal numeral forms (e.g. '15 savaitė' (15th week) ↦ '*penkiolikta* savaitė' (non-pronominal form) or '*penkioliktoji* savaitė' (pronominal form)). Other ambiguities were related to numeral case (e.g. '2019 vasarį' (2019 February) ↦ '*du tūkstančiai devynioliktųjų* vasarį' or '*du tūkstančiai devynioliktaisiais* vasarį'). All suffixes that represented a 'normalization hint' were eliminated from the data set (e.g. '2019-aisiais' was replaced by '2019'). This had an effect of making training subset of the corpus more interesting for the training algorithm, increased the complexity of the normalization task, and reduced the normalization accuracy estimates on the test subset of the corpus.

Sentences of the corpus were pre-processed by the Hidden Markov Models (HMM) based POS tagger [8]. Every text token was labeled with the so-called 'detailed' (or composite) morphological label that contained the following information:

- Lemma
- Part of speech (Noun, Verb, Adjective,...)
- Case (Nominative, Genitive,...)
- Gender (Feminine, Masculine)
- Number (Singular, Plural)

Since important prediction decisions are based on tagger provided POS tags and lemmas, to ensure optimal performance, morphological annotations were hand-corrected in training-testing data-set. When using morphological analysis data, it is beneficial to divide POS tags into sub-labels to build more abstract grammar rules and filter out redundant information.

B. Number grammar

Similar more saturated Natural Language Processing (NLP) sequence labeling tasks, POS tagging and Named Entity Recognition (NER), does not require hand-written language-specific grammar rules to achieve state-of-the-art [9] performance. Long-Short Term Memory neural networks coupled with Conditional Random Fields (LSTM-CRF) based data-driven sequence labeling approaches prove to be insufficient to achieve desired number normalization quality, considering training-data availability limitations. To efficiently leverage small data-set, morphosyntactic knowledge should be exploited for crafting language and task-specific grammar rules.

All rules are constructed as conditional functions without any prior weighting. Here are different techniques used to approximate and generalize number relationship with sentence context:

- Lemma classification. (Replacing certain word lemmas with a dedicated class name, for example, month names with '%Month')
- Number classification. (See Table I)
- Verb classification. (Based on syntactic features of controlling case of other POS)
- Syntactic linking (See Section V. Long-Distance Dependencies)

TABLE I. NUMBER CLASSIFICATION EXAMPLE

Num.	Roman	Int	Digit count	Req.Gen.*	Req.Sing.*
21	-	+	2	-	+
113.5	-	-	3	+	-
IV	+	+	1	-	-

*Requires Genitive and Requires singular signifies that countable noun of certain number must be of Genitive case or Singular

IV. MODELS

In this paper, mainly variations of Conditional Random Fields (CRF) are explored, since CRF got better baseline performance than its neural version (LSTM-CRF) and appears to be more suitable for particular data-set and grammar rule-set.

A. Sub-Label models

To create a single sequence tagging model for this task, we would need 79 (different combinations of sub-labels shown in Table II) detailed morphological labels corresponding to the output classes. With the currently available corpus, this would cause significant data scarcity problems. There are no training examples for a considerable number of classes and many others are barely represented. A good way to address this data scarcity problem is creating three independent CRF models for Case, Type, Gender prediction and to combine these predictions at a later stage. This is preferable since there is no direct dependency relationship between these morphological categories and operating with sub-labels allows creating a more abstract rule-set. Although, it is worth noting that sub-label dependencies have been proven useful for NLP sequence labeling using CRF [10] in combination with composite labels. Additionally exploiting composite label dependencies might be beneficial for number normalization as well, and it is worth exploring in future research.

TABLE II. MORPHOLOGICAL SUB-LABELS

Case	Type/Number	Gender
Nominative	Cardinal	Feminine
Genitive	Ordinal singular	Masculine
Dative	Ordinal plural	Not applicable
Accusative	Ordinal definitive singular	
Instrumental	Ordinal definitive plural	
Locative	Cardinal multiple	
Not applicable	Month*	

*This class is designed for a number that could be substituted with month name, for example, '2019-02-03' and '2019-February-03'

B. Skip-Chain CRF

The linear-chain structure is usually used for sequence labeling CRF since additionally modeling non-linear relationships requires complicated inference algorithms and prior specification of such dependencies [11], [12]. For number normalization, a simplified version of Skip-chain Conditional Random Field (S-CRF) can be used, as shown in the gender prediction model comparison in Figure 1 and Figure 2 (for readability reasons, we only show English glossary sentence example of Lithuanian model). Both graphs are representations of Viterbi algorithm decoding (the same structure is used for encoding). Circles correspond to nodes and arrows to transitions. The weight of node or transition is calculated as the sum of its conditional feature-set weights (unigrams for node and bigrams for transition). 'f' and 'm' are notations for 'feminine' and 'masculine' genders, while '0' represents a class for non-number tokens, which are not to be changed by the normalization task. The blue path is the correct path selected by the Viterbi algorithm. In Linear-chain CRF (L-CRF) most bigram features are useless since they connect with non-number tokens (in Figure 1 none of transition weights are significant). This means we effectively have zeroth-order CRF. Transitions that are actually important are between numbers. To implement such dependencies we make two sequences - full (original) and skip (numbers only). We build unigram features only for number tokens but from the full sequence. This way our unigram features exactly match those of the linear-chain model. Next, bigram features are built from the skip sequence. For encoding and decoding, we use skip sequence as well, since we do not build any feature functions for non-number tokens. Skip-chain models, as described above, have unaltered

unigram and improved bigram function sets (for number tokens), while being significantly faster (see graph simplification shown in Figure 1 and Figure 2). Our implementation uses a modified version of the CRFSharp toolkit [13].

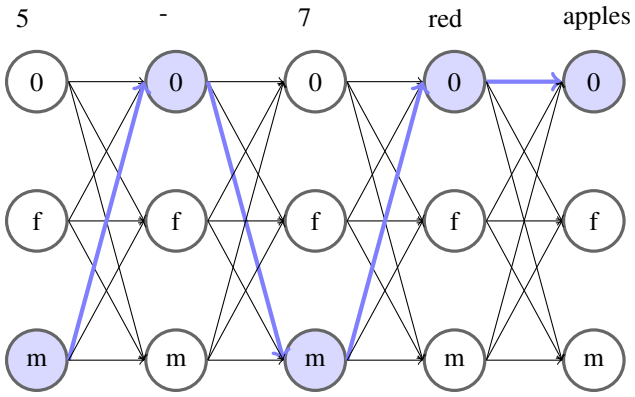

Fig. 1. First-Order Linear-Chain CRF

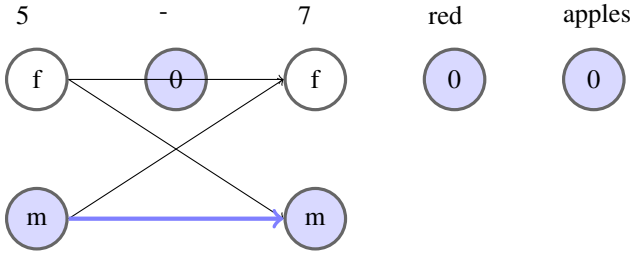

Fig. 2. First-Order Skip-Chain CRF

V. Long-Distance Dependencies

Although skip-chain structure quite reliably models some important long-distance relationships, it is not able to capture distant dependencies between number and non-number tokens (e.g. in '3 didžiųjų mobiliojo ryšio operatorių' $\mapsto$ '*trijų* mobiliojo ryšio operatorių' (three major mobile network operators) numbers '3' case is determined by words 'operatorių' (operators) case). CRF is generally unable to leverage such features and requires either hybridization such as LSTM-CRF, or additional pre-processing. We propose identifying position-distance independent relationships using an ad-hoc set of linkage rules and formulating perceived syntactic links as conditional functions of CRF. For Lithuanian language, we discern three directly related parts of speech (Noun, Verb, Preposition) in the number normalization task. For each, we use a different set of linkage rules, to identify related tokens to every number in the sentence, effectively performing partial syntactic analysis. To link prepositions and verbs to numbers, our rules solely rely on morphological labels provided by the POS tagger. For nouns, the task of linking could be more precisely formulated as an identification of a noun which represents an object or quantity being counted by some number in the sentence. This is extremely important since countable noun has crucial morphological information. For example, with successful identification, we no longer need to predict the gender of a numeral, since it is directly determined by the noun.

A. Countable noun identification

We determine the most likely countable noun in a two-step process. First, for a given numeric token d we select all potentially countable noun tokens $\{n_i\}$ according to the ad-hoc set of linkage rules for nouns. We can not make an educated choice among selected nouns on the basis of available morphosyntactic annotation since noun morphology does not have the property of 'countability'. To discriminate among potential countable nouns, semantic analysis is needed. We need to rate the set of selected nouns $\{n_i\}$ according to some 'countability' measure ξ that is dependent on the numeric token d being normalized and select the noun n_{best} with the highest $\xi(d, n_i)$ rating $n_{best} = \arg\max \xi(d, n_i)$

Suppose that we have vector embeddings $v(n), v(n) \in R_D$ for every noun n, that were obtained by an algorithm such as 'word2vec' [14]. Suppose that we also designed a mapping ϕ that maps every numeric token d into a vector $\phi(d), \phi(d) \in R_D$ such that $\phi(d)$ is the representative embedding of the set of nouns that are frequently counted by the numeric token d. If both assumptions are correct, we can rate the set of potential countable nouns by estimating cosine similarity between each selected noun and the corresponding representative vectors, i.e.

$$\xi(d, n_i) = \text{cosine-similarity}(\phi(d), v(n_i)) \quad (1)$$

We have tested a few different approaches to design the above-mentioned mapping $\phi(d)$. We sought large unannotated text corpus for number and noun adjacent co-occurrences and made noun frequency lists per every numeric token that was found (around 350 thousand co-occurrences). Information present in a frequency list can be aggregated into a single vector by estimating the weighted average of noun embeddings making up that list. Thus a representative (or central) embedding vector can be obtained per every numeric token. Although this tabular mapping from numeric tokens into representative vectors can be used in (1), it has serious limitations. The table contains many unreliable vectors for rare numbers, because of the lack of co-occurrences in the unannotated corpus. To circumvent the limitations of this tabular mapping we used the Neural Network (NN) approach to build a continuous co-occurrence model. We built two different neural networks: one with a single input (corresponding to the mathematical value of the numeric token) and one with 7 inputs, corresponding to the decomposition of the numeric token into sub-parts (thousands, hundreds,...) and including number features similar to Table I. NN had 200 output units.

The evaluation of these models is shown in Table III. The baseline performance is obtained by the simple rule "take the first potentially countable noun to the right of a numeric token". Accuracy is measured using whole CRF training data, extracting situations where a choice between two or more nouns (2.41 avg.) is needed.

TABLE III. COUNTABLE NOUN LINKING

Method	Accuracy
Select first	68.77
1-input NN	84.11
7-input deep NN	87.40

VI. EVALUATION

We evaluate models with 5-fold cross-validation (except for countable noun identification in Section V, since training and testing data-sets were obtained from different sources). The accuracy of different models is shown in Table IV. Combined accuracy estimates the accuracy of all three models. The combined answer is considered to be correct if all three sub-labels are correct.

It is worth noting, that our model is focused on grammatically correct, as 'spoken' number normalization. This might not be desirable for systems like text-to-speech synthesis, hence a more standardized approach can be chosen. For Lithuanian language, numeral definiteness property could be removed from the prediction model, since it is not strictly constrained by grammar. This would increase language correctness and improve Type prediction models and combined accuracy as shown last line of Table IV (best performing model without definiteness property).

Accuracies above represent the lower bound accuracies of the real-world number normalization performance. Firstly, in certain situations, some sub-label prediction mistakes might be irrelevant for numeral generation. For example, both '5, Cardinal, Genitive, Feminine' and '5, Cardinal, Genitive, Masculine' will generate the same word representation 'penkių'. Secondly, real-world sentences often contain suffixes (e.g. 'Kovo 11-ąją' $\mapsto$ 'Kovo *vienuoliktąją*' (March 11th)) that either offer an unambiguous hint that solves the number normalization problem or at least provides most of the needed morphological information, which can be used to correct prediction mistakes.

TABLE IV. EVALUATION

	Case	Type	Gender	Combined
L-CRF	77.19	89.00	94.79	67.52
S-CRF	78.89	89.82	95.17	69.43
S-CRF+class.*	83.81	93.64	95.17	76.49
S-CRF+class.+syn.**	86.05	94.01	98.51	80.91
without Definiteness	86.05	96.82	98.51	83.08

*classification, see Section III. Preprocessing
**syntactic analysis, see Section V. Long-distance dependencies

VII. CONCLUSIONS

In this paper, we describe the number normalization disambiguation model, which is needed to develop a context dependant number-to-words system. Sequence-labeling approach allows us to normalize countable abbreviations and symbols (next to number) effortlessly, since countable noun morphological form can be extracted from predicted label (e.g. 'nuo 5%' (from 5%) $\mapsto$ 'nuo *penkių procentų*'). Our implementation based on this model is publicly available [15] and in the future will be integrated into a full Lithuanian text normalization system.

Number normalization errors are often directly dependant on morphological analysis mistakes and we are currently working on improving both vocabulary-grammar and disambiguation sides of Lithuanian POS tagging to consequentially increase number normalization accuracy.

Currently, we use 'word2vec' [14] algorithm trained on relatively small text corpus to produce word embeddings. Although various improvements have been made in encoding text semantic information to vectors [16], [17] and using more advanced method and larger corpus would likely improve our model performance.

Our achieved number normalization accuracy could be further improved by expanding annotated training data since a considerable amount of errors are a direct result of data scarcity. Although, our approach generally lacks semantic and syntactic language understanding, so performing full syntactic sentence analysis in the preprocessing stage would be highly beneficial.

ACKNOWLEDGMENT

This research was supported by the project "Semantika 2" (No. 02.3.1-CPVA-V-527-01-0002). Special gratitude goes to our colleagues Lina Majauskaitė and Dovilė Stukaitė who helped us in collecting and annotating text corpus.

REFERENCES

[1] V. Dadurkevičius. dadurka/number-to-words-lt. [Online]. Available: https://github.com/dadurka/number-to-words-lt
[2] K. Wu, K. Gorman, and R. Sproat. (2016) Minimally supervised written-to-spoken text normalization.
[3] M. Wróbel, J. T. Starczewski, and C. Napoli, "Handwriting recognition with extraction of letter fragments," in *International Conference on Artificial Intelligence and Soft Computing*. Springer, 2017, pp. 183–192.
[4] R. Sproat and N. Jaitly. (2016) Rnn approaches to text normalization: A challenge.
[5] T. Kapuściński, R. K. Nowicki, and C. Napoli, "Comparison of effectiveness of multi-objective genetic algorithms in optimization of invertible s-boxes," in *International Conference on Artificial Intelligence and Soft Computing*. Springer, 2017, pp. 466–476.
[6] K. Gorman and R. Sproat, "Minimally supervised number normalization," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 507–519, 2016. [Online]. Available: https://www.transacl.org/ojs/index.php/tacl/article/view/897/213
[7] T. Kapuściński, R. K. Nowicki, and C. Napoli, "Application of genetic algorithms in the construction of invertible substitution boxes," in *International Conference on Artificial Intelligence and Soft Computing*. Springer, 2016, pp. 380–391.
[8] [Online]. Available: http://donelaitis.vdu.lt/main_helper.php?id=4&nr=7_2
[9] A. Akbik, D. Blythe, and R. Vollgraf, "Contextual string embeddings for sequence labeling," in *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, 2018, pp. 1638–1649. [Online]. Available: http://aclweb.org/anthology/C18-1139

[10] M. Silfverberg, T. Ruokolainen, K. Lindén, and M. Kurimo, "Part-of-speech tagging using conditional random fields: Exploiting sub-label dependencies for improved accuracy," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, 2014, pp. 259–264. [Online]. Available: http://aclweb.org/anthology/P14-2043

[11] M. Galley, "A skip-chain conditional random field for ranking meeting utterances by importance," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '06. Stroudsburg, PA, USA: Association for Computational Linguistics, 2006, pp. 364–372. [Online]. Available: http://dl.acm.org/citation.cfm?id=1610075.1610126

[12] J. Liu, M. Huang, and X. Zhu, "Recognizing biomedical named entities using skip-chain conditional random fields," in *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, ser. BioNLP '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 10–18. [Online]. Available: http://dl.acm.org/citation.cfm?id=1869961.1869963

[13] Z. Fu. zhongkaifu/crfsharp. [Online]. Available: https://github.com/zhongkaifu/CRFSharp

[14] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781, 2013. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1301.html#abs-1301-3781

[15] [Online]. Available: http://prn509.vdu.lt:9080/

[16] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[17] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *In EMNLP*, 2014.