Assessing the Importance of Frequency versus Compositionality for Subword-based Tokenization in NMT

Benoist Wolleb, Romain Silvestri, Giorgos Vernikos, Ljiljana Dolamic and Andrei Popescu-Belis

HEIG-VD

EPFL

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

armasuisse

Motivation

BPE has three advantages: short encoding of frequent words, subword compositionality, unknown words

Which one is more important ?

Idea: Alternative tokenization method based on another compression algorithm without compositionality

Method: Build *n*-ary Huffman tree and use "compression" symbols to tokenize words

Data

Corpora
News Commentary
Europarl
Common Crawl
JW300
Newstest

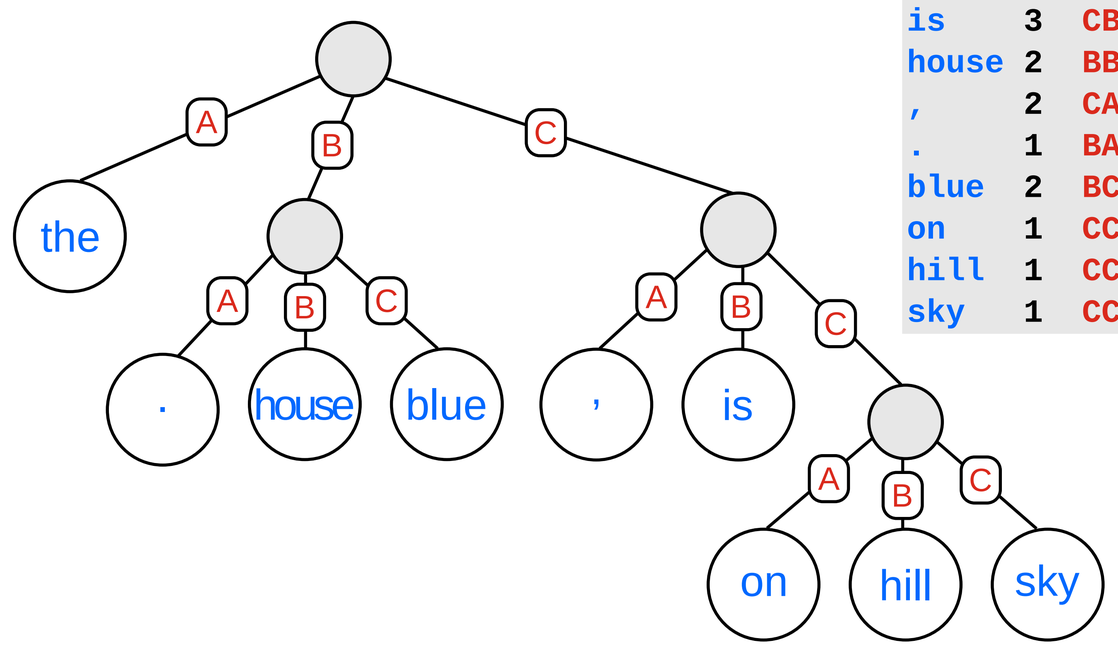
	Lines	
Languages	**Train**	**Test**
CS - DE	1'780'068	3444
EN - DE	4'547'445	4615
EN - FR	5'245'392	4448

Example

the house is on the hill, the house is blue, the sky is blue.

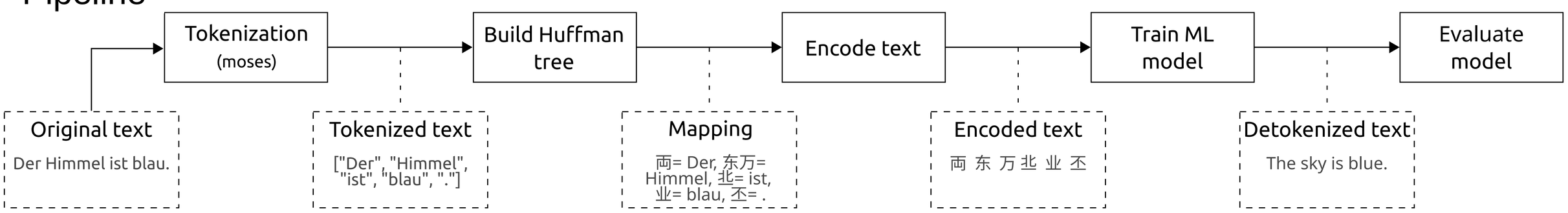
n = 3

type	nb	code
the	4	A
is	3	CB
house	2	BB
,	2	CA
.	1	BA
blue	2	BC
on	1	CCA
hill	1	CCB
sky	1	CCC



A BB CB CCA A CCB CA A BB CB BC CA A CCC CB BC BA

Pipeline



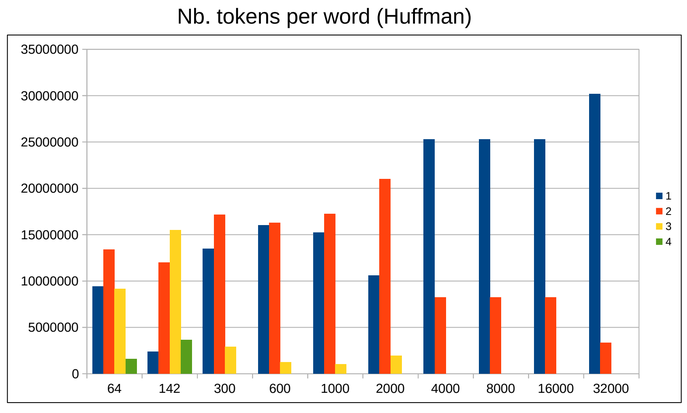
Algorithm

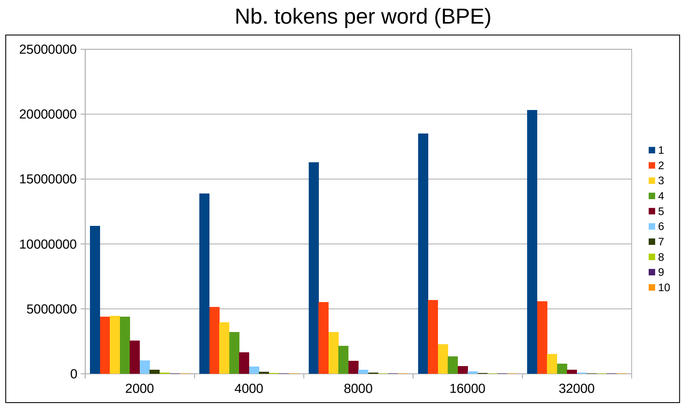
Data: Word frequencies F: $\{(w_i, f_i), \ldots\}$,
Priority queue H: $\{(node_i, score_i), \ldots\}$ sorted by increasing scores,
Number of symbols: n
Result: Huffman tree
foreach $(w_i, f_i) \in F$ **do**
  Create $node_i$ with key w_i and score f_i;
  Add $node_i$ to H;
end
while *length*$(H) > 1$ **do**
  $L \leftarrow$ empty list of nodes;
  $S \leftarrow 0$;
  for $i \leftarrow 0$ **to** n **do**
    if $H = \emptyset$ **then**
      break;
    else
      Pop $(node_i, score_i)$ from H;
      Append $(node_i, score_i)$ to L;
      Add $score_i$ to S;
    end
  end
  Create new node N = ('None', S);
  foreach $node \in L$ **do**
    Add $node$ to N's children;
  end
  Push N to H;
end

Results

Lang. pair	Nb. of symbols	BLEU			ChrF			COMET		
		Huffman	BPE	%	Huffman	BPE	%	Huffman	BPE	%
CS-DE	2k	20.3	24.4	83.2	46.6	52.6	88.6	0.758	0.829	91.4
	4k	20.9	24.8	84.3	47.2	53.2	88.7	0.762	0.833	91.4
	8k	21.6	25.1	86.1	48.4	53.4	90.6	0.780	0.834	93.6
	16k	22.3	24.8	89.9	49.3	53.3	92.5	0.791	0.830	95.2
	32k	23.1	26.4	87.5	50.2	54.5	92.1	0.804	0.837	96.0
EN-DE	8k	19.5	22.4	87.1	46.4	49.7	93.4	0.709	0.769	92.2
	16k	20.3	22.2	91.4	46.6	49.3	94.5	0.718	0.768	93.5
	32k	19.8	22.5	88.0	46.9	49.5	94.7	0.712	0.772	92.2
EN-FR	8k	27.1	31.2	86.9	51.1	55.3	92.4	0.728	0.783	93.0
	16k	27.6	30.9	89.3	51.8	55	94.2	0.739	0.781	94.6
	32k	27.9	30.9	90.3	52.2	54.9	95.1	0.746	0.784	95.1

Tokenization





Huffman coding uses at most 4 symbols per token, BPE uses up to 10

Findings

Translation quality does not decrease much with Huffman
Compositionality is not the most important aspect of suwbords.

Frequency is the major factor, contributing for

- 90.2 % of BLEU
- 93.7 % of ChrF
- 94.4 % of COMET

[1] https://github.com/heig-iict-ida/huffman-tokenizer
[2] Sennrich et al. 2016. Neural machine translation of rare words with subword units. *Proc. of ACL.*
[3] Chitnis et al. 2015. Variable-length word encodings for neural translation models. *Proc. of EMNLP.*
[4] Huffman. 1952. A method for the construction of minimum-redundancy codes. *Proc. of IRE.*
[5] Kudo et al. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *Proc. of EMNLP.*