Compositional augmentation policy using different formulas for the notion of middle sentence for low resource machine translation

CHEN Jin and Yves LEPAGE
早稲田大学大学院情報生産システム研究科
jin.chen@asagi.waseda.jp yves.lepage@waseda.jp

Abstract

Middle sentence generation is a technique that, given a start sentence and an end sentence, outputs a sentence with middle semantics. We aim to further explore formulas for middle sentence generation and construct a compositional policy to combine these formulas into better translation models. We study particular and general types of formulas: particular formulas are based on specific words and general formulas are based on the IDF score of each individual word. We use these formulas as corpus augmentation operations and define a policy that automatically constructs a breadth-first tree which finds the node with the least perplexity as the best node. Results show that our policy provides significant improvements over our baseline.

1 Introduction

The idea of middle sentence generation is based on a specific analogy [1]:

Start : Middle :: Middle : End

It takes two sentences from the corpus as start sentence and end sentences and generates a new sentence whose semantics lies at the semantic midpoint of the two start and end sentences. If sentences are represented by vectors, the basic formula for the vector of a middle sentence is thus as follows:

$$\overrightarrow{\text{Middle}}_{\text{basic}} = \frac{1}{2} \times (\overrightarrow{\text{Start}} + \overrightarrow{\text{End}}) \tag{1}$$

Some studies have demonstrated the effectiveness of data augmentation methods based on the idea of middle sentence generation for natural language processing tasks [2, 3, 4, 5, 6]. However, they have limitations as they explore various different formulas by evaluating various data augmentation results that use these different formulas.

This paper aims to continue the exploration of generation formulas that use the notion of middle sentence, by constructing a comprehensive framework that builds upon several proposed generation formulas.

2 Sentence generation based on the notion of middle sentence

2.1 Basic and renormalized formulas

New sentences can be generated directly as middle sentences. This has been explored in works like [4]. The main drawback is that the middle vector might not be a vector that actually fits a sentence vector. The consequence is that, when decoding from the vector, the generated sentence suffers problems: repeated words, sentences somehow falling short off, inconsistent semantics.

So as to remedy to this problem and release the constraint on the middle vector, more freedom can be given to generation by creating a vector sustained by the start and the middle sentence, instead of the start and end sentence [6]. This allows the creation of sentences with possibly new semantics.

$$\overrightarrow{\text{End}}_{\text{basic}} = 2 \times \overrightarrow{\text{Middle}} - \overrightarrow{\text{Start}} \tag{2}$$

The previous formulas (1) and (2) do not take into account the fact that the norms of the generated middle or end vectors might be too short for being an accurate vector representation of a sentence. Renormalization is an answer to this problem. It makes the vectors easier to decode into reasonable sentences by reducing the non-felicitous influence of too short norms [4, 5, 6]. The renormalization formulas are:

$$\overrightarrow{\text{Middle}}_{\text{renorm}} = \frac{||\overrightarrow{\text{Start}}|| + ||\overrightarrow{\text{End}}||}{||\overrightarrow{\text{Start}} + \overrightarrow{\text{End}}||} \times \overrightarrow{\text{Middle}}_{\text{basic}} \tag{3}$$

$$\overrightarrow{\text{End}}_{\text{renorm}} = \frac{2 \times ||\overrightarrow{\text{Middle}}|| - ||\overrightarrow{\text{Start}}||}{||2 \times \overrightarrow{\text{Middle}} - \overrightarrow{\text{Start}}||} \times \overrightarrow{\text{End}}_{\text{basic}} \quad (4)$$

In this paper, we adopt generation of the end sentence.

2.2 Influence of specific words

Taking into account low-frequency words can provide richer and more acute semantics to the generated vector representations. For the same reasons, giving some weight to hapaxes can lead to vectors positively biased towards rarer formulations. By contrast, inhibiting the influence of more frequent words can solve the problem of overloading the generated sentence with grammatical words. For all the previous reasons, we propose a series of formulas that leverage the frequency of different words.

2.2.1 Low frequency words

It has been observed that giving more weight to rare words can enrich the calculated sentence vectors and can lead to better generated sentences [4]. The low frequency words in appearing in the two put sentences can be added to the sentence vector weighted by some factor λ. Although the original proposal was for middle sentences, we apply it to end sentences:

$$\overrightarrow{\text{End}}_{\text{lowfreq}} = \overrightarrow{\text{End}}_{\text{renorm}} + \lambda \times \overrightarrow{\text{word}}_{\text{lowfreq}} \quad (5)$$

2.2.2 Hapaxes

A reinforced view over the previous one that took into account low frequency words, is to consider the addition of the average of all hapax embeddings in the start and middle sentences.

$$\overrightarrow{\text{End}}_{\text{hapaxes}} = \frac{1}{1+n} \times \left(\overrightarrow{\text{End}}_{\text{basic}} + \lambda \times \sum_{1}^{n} \overrightarrow{\text{word}}_{\text{hapaxes}}\right) \quad (6)$$

2.2.3 Stop words

The previous formula for hapaxes added information to the sentence vector. Similarly, and by opposition, one can imagine subtracting the average vector of all embeddings of stop words to reduce their influence on the generated sentence.

$$\overrightarrow{\text{End}}_{\text{stopwords}} = \overrightarrow{\text{End}}_{\text{renorm}} - (\lambda \times \frac{1}{n} \sum_{1}^{n} \overrightarrow{\text{word}}_{\text{stopwords}}) \quad (7)$$

2.3 Generalization: taking idf of each word into account

A generalization of the previous formula can be obntained by taking into account their inverse document frequecy (idf). Here, documents are each individual sentence. This allows to balance the information of words according to tehir frequency. The additional part is adapted through the factor λ.

$$\overrightarrow{\text{End}}_{\text{all}} = \overrightarrow{\text{End}}_{\text{renorm}} - \left(\lambda \times \frac{1}{n} \sum_{1}^{n} (\text{idf} \times \overrightarrow{\text{word}})\right) \quad (8)$$

In order to not influence too much by the norm of each word, this last formula renormalizes words before incorporating their embedding to the calculated vector.

$$\overrightarrow{\text{End}}_{\text{all+norm}} = \overrightarrow{\text{End}}_{\text{renorm}} - \left(\lambda \times \frac{1}{n} \sum_{1}^{n} (\frac{\text{idf}}{||\overrightarrow{\text{word}}||} \times \overrightarrow{\text{word}})\right) \quad (9)$$

3 Compositional augmentation policy

Text AutoAugment is a compositional framework whose core idea is to generate new synthetic text by combining individual editing operations to form a complete sequence [7]. The framework is not only capable of generating multiple sequence instances at once, but it can also automatically select the best-performing augmented dataset among all the ones generated. Text AutoAugment has been proposed for text classification.

We propose to adopt this compositional framework for data augmentation for the task of data augmentation in machine translation. The operations in our framwork will be the proposed formulas for sentence generation. Our framework defines a similar policy to Text AutoAugment, called $\mathcal{P}$, that contains N operations $\mathfrak{O}$. For each operation $\mathfrak{O}$, we define:

$$\mathfrak{O} = \langle t, \lambda, \theta \rangle$$

where:

1. t is the type of generation formula, i.e. the five proposed formulas and the renormalized end sentence formula.
2. λ is the weight of the extra term in the formula. Note that the renormalized end sentence formula can be seen as the multiplication of λ with a zero vector. $\lambda \in [0, 1]$.

3. θ represents the threshold for sentence filtration. We use Euclidean distance to determine whether the bilingual sentences are parallel. $\theta \in [0.3, 0.6]$.

Our framework takes a parallel corpus as input, stores it to the root node of an exploration tree, and trains a machine translation model without any augmented data from the original parallel corpus. Subsequently, for each node, the framework creates N daughter nodes according to the number of operations in the policy $\mathscr{P}$. The corpus on a daughter node is obtained by data augmentation using the corresponding operation $\mathfrak{S}$.

At a tree depth of 1, the node's model is trained on the corresponding corpus while at depths greater than 1, the node's model is fine-tuned on the parent node's model using its own corpus. When the child node's perplexity is less than the parent node's, the end sentence generation of daughter nodes stops and this daughter node is discarded.

The framework will finally compare perplexities of all models on the leaves and select the node with the smallest perplexity as the best node, with the path from the root node to that node. This path represents the best operational solution. The complete framework is illustrated in Figure 1.

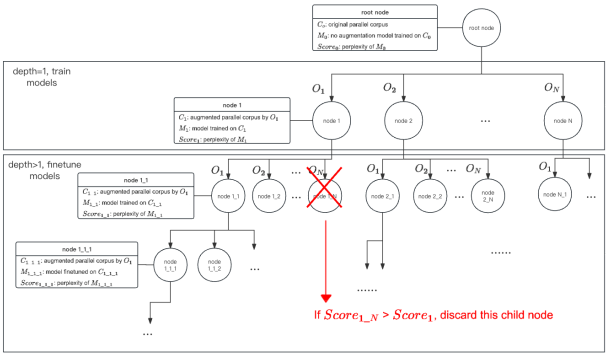
root node
C_o: original parallel corpus
M_0: no augmentation model trained on C_0
$Score_0$: perplexity of M_0
root node
depth=1, train models
O_1
O_2
O_N
node 1
C_1: augmented parallel corpus by O_1
M_1: model trained on C_1
$Score_1$: perplexity of M_1
node 1
node 2
...
node N
depth>1, finetune models
node 1_1
C_{1_1}: augmented parallel corpus by O_1
M_{1_1}: model trained on C_{1_1}
$Score_{1\_1}$: perplexity of M_{1_1}
O_1
O_2
...
O_N
node 1_1
node 1_2
node 1_N
O_1
O_2
O_N
node 2_1
node 2_2
node 2_N
O_1
node N_1
node 1_1_1
$C_{1\_1\_1}$: augmented parallel corpus by O_1
$M_{1\_1\_1}$: model finetuned on $C_{1\_1\_1}$
$Score_{1\_1\_1}$: perplexity of $M_{1\_1\_1}$
node 1_1_1
node 1_1_2
......
If $Score_{1\_N}$ > $Score_1$, discard this child node

Figure 1 Proposal: compositional augmentation framework

4 Experimental setup

4.1 Machine translation engine

We use the OpenNMT-py toolkit [8] to create all the necessary models. Each model is built using an encoder-decoder architecture based on the Transformer model [9].

4.2 Data

We use the German and Upper Sorbian language datasets from the WMT22 website[1] for our experiments. Our model training scheme follows the original division: 60,000 instances for training, 2,000 for validation, and 2,000 for testing. Details of data are shown in Table 1.

Lang.	# of sent.	Avg. # words/sent.	Vocab. size
Sorbian	64,000	12.32 ± 7.00	75,558
German		13.71 ± 7.49	55,387

Table 1 Statistics on WMT 2022 German–Upper Sorbian

5 Results

5.1 Analysis of different thresholds

We obtain 120,000 parallel sentence pairs when generating new sentences according to formulas. We set thresholds from 0.3 to 0.6 in steps of 0.05 for filtration and conduct seven series of experiments with the best weight of each formula. BLEU scores of models trained based on the augmented corpora generated by each formula at different thresholds for German to Upper Sorbian translation are shown in Figure 2.

Overall, the increase in the threshold value has a negative impact on the model, leading to a decrease in BLEU, but the threshold does not allow a more refined representation of the differences in the augmented corpus. Most models perform worst at a threshold of 0.6 as well as best at a threshold of 0.4. The best model uses the renormalized idf-weighted formula with a threshold of 0.4.

Comparing the best results for each formula, all models trained from formulas perform better than the unaugmented model and none of the formulas except the formula for averaged hapaxes scores perform lower than the baseline in BLEU. The renormalized idf-weighted formula, that was proposed last as a generalization (see Formula (9)), shows significant improvement in terms of BLEU and chrF2. Both the IDF-weighted formula and the renormalized IDF-weighted formula show relatively superior performance in both directions.

[1] EMNLP 2022 seventh conference on machine translation (WMT22)

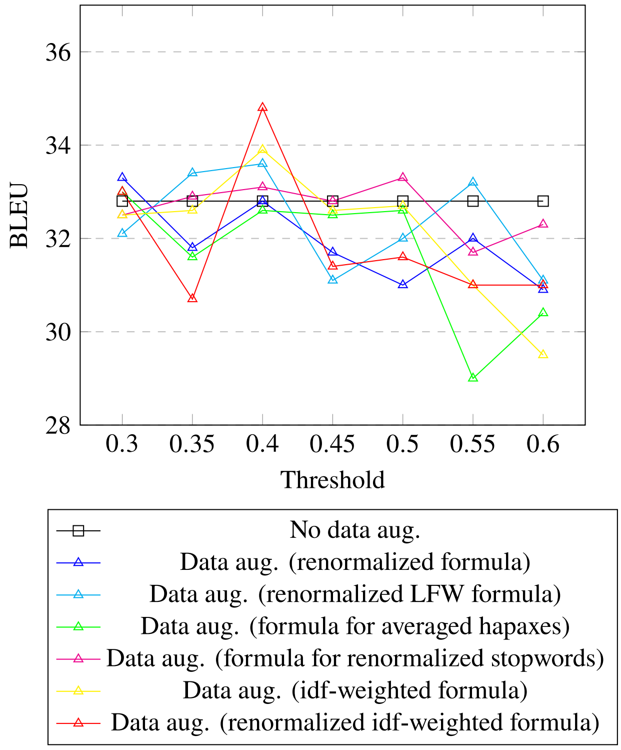

Figure 2 Variation in BLEU scores when using different thresholds (German to Upper Sorbian)

5.2 Compositional augmentation policy

We construct a policy with N=6 operations. Each operation is a separate formula with its optimal weight and threshold.

We compare the optimal nodes in both directions and compare with the non augmented model and the baseline model. Our policy shows significant improvements in all three metrics. This is shown in Table 2.

Direction	Method	# of add. sent.	BLEU	chrF2	TER
de → hsb	No aug.	-	32.8 ± 1.2	58.3 ± 0.8	44.5 ± 1.0
	Renorm.	116	33.3 ± 1.1	59.0 ± 0.8	43.7 ± 1.0
	Com. policy	1,840	**35.0 ± 1.1**	**64.9 ± 0.8**	**40.9 ± 0.9**
hsb → de	No aug.	-	32.5 ± 1.1	60.2 ± 0.8	46.7 ± 1.0
	Renorm.	1,303	33.4 ± 1.1	61.1 ± 0.8	46.1 ± 1.0
	Com. policy	1,213	33.3 ± 1.1	**63.7 ± 0.8**	**41.8 ± 0.9**

Table 2 Comparison of compositional policy with no augmentation model and baseline model

The results in Table 3 show details of our policy in the direction German to Upper Sorbian. We observe that fine-tuned nodes at depth 2 all have much lower perplexity than the first layer of the trained-only model, while no node arrives at depth 3. The main reason is that the amount of newly generated data in this low-resource setting is insufficient.

Depth	Node No.	Parent Node No.	Total path (s)	Perplex.	Leaf node
0	0	-	-	19.5764	False
1	1	0	Avg. Hapax	19.2489	False
	2		IDF-weig.	18.8499	False
	3		Re. IDF-weig.	18.7726	False
	4		Re. LFW	18.5069	False
	5		Renorm.	19.2353	False
	6		Re. Stop.	19.2027	False
2	7	1	Avg. Hapax + Avg. Hapax	6.4694	True
	8		Avg. Hapax + IDF-weig.	6.4013	True
	9		Avg. Hapax + Re. IDF-weig.	6.5311	True
	10		Avg. Hapax + Re. LFW	6.4530	True
	11		Avg. Hapax + Renorm.	6.5018	True
	12		Avg. Hapax + Re. Stop.	6.4304	True
	13	2	IDF-weig. + IDF-weig.	6.4736	True
	14		IDF-weig. + Re. IDF-weig.	6.4891	True
	15		IDF-weig. + Re. LFW	6.4090	True
	16		IDF-weig. + Renorm.	6.4590	True
	17		IDF-weig. + Re. Stop	6.3949	True
	18	3	Re. IDF-weig. + Avg. Hapax	6.3812	True
	19		Re. IDF-weig. + IDF-weig.	6.4937	True
	20		Re. IDF-weig. + Re. IDF-weig.	6.4575	True
	21		Re. IDF-weig. + Re. LFW	6.3896	True
	22		Re. IDF-weig. + Renorm.	6.3787	True
	23		Re. IDF-weig. + Re. Stop.	6.4301	True
	24	4	Re. LFW + Avg. Hapax	6.3645	True
	25		Re. LFW + IDF-weig.	6.4475	True
	26		Re. LFW + Re. IDF-weig.	6.4883	True
	27		Re. LFW + Re. LFW	6.4563	True
	28		Re. LFW + Renorm.	6.4937	True
	29		Re. LFW + Re. Stop.	6.4613	True
	30	5	Renorm. + Avg. Hapax	6.3420	True
	31		Renorm. + IDF-weig.	6.3985	True
	32		Renorm. + Re. IDF-weig.	6.3734	True
	33		Renorm. + Re. LFW	6.4433	True
	34		Renorm. + Re. Stop.	6.3937	True
	35	6	Re. Stop. + IDF-weig.	6.3157	True
	36		Re. Stop. + Re. IDF-weig.	6.4081	True
	37		Re. Stop. + Re. LFW	6.3522	True
	38		Re. Stop. + Renorm.	6.3310	True
	39		Re. Stop. + Re. Stop.	6.3792	True

Table 3 Results of nodes in compositional policy (German to Upper Sorbian). Node with least perplexity is No. 35, generation path is formula for renormalized stopwords + idf-weighted formula.

6 Conclusion and limitations

Our results demonstrated significant improvements with our model: +1.7 in BLEU, +5.9 in chrF2, and -2.8 reduction in TER in the German to Upper Sorbian direction. In the Upper Sorbian to German direction, the improvements obtained are: +2.6 in chrF2 and -4.3 in TER.

As a limitation, with our policy, the exploration tree is difficult to generate at higher depths in this low-resource setting. Also, the cost of constructing a compositional augmentation tree is high, which leads to the issue of local optima when determining hyper-parameters.

References

[1] Pengjie Wang, Liyan Wang, and Yves Lepage. Generating the middle sentence of two sentences using pre-trained models: a first step for text morphing. In **Proceedings of the 27th annual meeting of the Association for Natural Language Processing**, pp. 1481–1485, 2021.

[2] Koki Osawa and Yves Lepage. Data augmentation of parallel data for style transfer by generation of middle sentences (in japanese). In **Proceedings of the 28th Annual Conference of the Association for Natural Language Processing**, pp. 1376–1380, Hamamatsu, Japan, March 2021. Association for Natural Language Processing.

[3] Zhicheng Pan, Xinbo Zhao, and Yves Lepage. Sentence analogies for text morphing. In **Proceedings of the workshop 'Analogies: from Theory to Applications (ATA@ICCBR 2022)', held in conjunction with the 30th International Conference on Case-Based Reasoning (ICCBR)**, Nancy, France, 2022.

[4] Matthew Eget, Xuchen Yang, and Yves Lepage. A study in the generation of multilingually aligned middle sentences. In Zygmunt Vetulani and Patrick Paroubek, editors, **Proceedings of the 10th Language & Technology Conference (LTC 2023) – Human Language Technologies as a Challenge for Computer Science and Linguistics**, pp. 45–49, April 2023.

[5] Wenyi Tang and Yves Lepage. A dual reinforcement method for data augmentation using middle sentences for machine translation. In Masao Utiyama and Rui Wang (Research Track Co-Chairs), editors, **Proceedings of Machine Translation Summit XIX**, Vol. 1: Research Track, pp. 48–58, Macau SAR, China, September 2023.

[6] Xiyuan Chen. Data augmentation for machine translation using the notion of middle sentences. Master's thesis, Graduate School of Information, Production and Systems, Waseda University, 2023.

[7] Shuhuai Ren, Jinchao Zhang, Lei Li, Xu Sun, and Jie Zhou. Text AutoAugment: Learning compositional augmentation policy for text classification. In **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing**, pp. 9029–9043, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[8] Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. OpenNMT: Open-source toolkit for neural machine translation. In **Proceedings of ACL 2017, System Demonstrations**, pp. 67–72, Vancouver, Canada, 2017. Association for Computational Linguistics.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, **Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA**, pp. 5998–6008, 2017.

A Result analysis under different thresholds (the other direction)

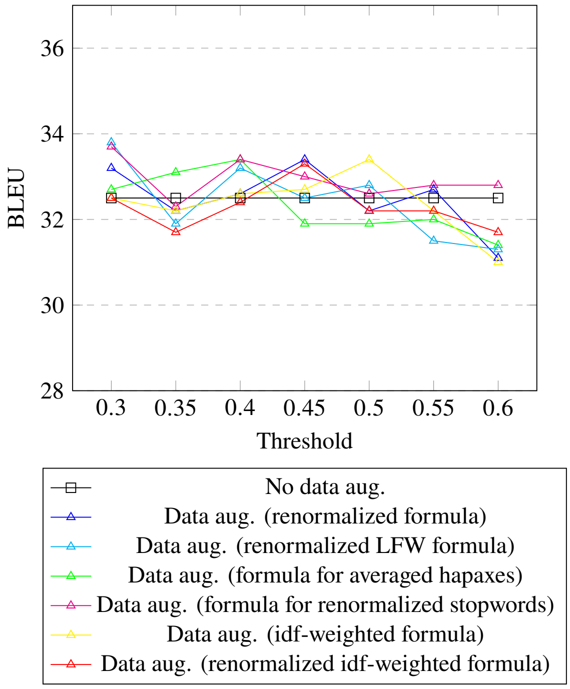

Figure 3 Same as Figure 2, but for the other direction (Upper Sorbian to German)

B Comparison of formulas under optimal parameters

We give the statistics of the results of models trained under each formula. Results are shown in Tables 4 and 5.

Formula	Weig. (λ)	Thre. (θ)	# of added sent.	BLEU	chrF2	TER
- (no aug.)	-	-	-	32.8 ± 1.2	58.3 ± 0.8	44.5 ± 1.0
Renorm.	-	0.30	116	33.3 ± 1.1	59.0 ± 0.8	43.7 ± 1.0
Renorm. LFW	0.1	0.40	745	33.6 ± 1.1	59.2 ± 0.8	43.8 ± 1.0
Avg. hapaxes	0.5	0.30	212	33.0 ± 1.1	58.3 ± 0.9	45.1 ± 1.0
Renorm. Stop.	0.4	0.50	1,210	33.3 ± 1.1	59.0 ± 0.8	44.2 ± 0.9
IDF-weig.	0.9	0.40	511	33.9 ± 1.1	59.2 ± 0.8	43.7 ± 0.9
Renorm. IDF-weig.	0.9	0.40	380	**34.8 ± 1.2**	**60.1 ± 0.8**	43.0 ± 0.9

Table 4 Statistics of best results for each formula (German to Upper Sorbian). Scores in bold face are statistically different and higher than the baseline (no aug.).

C Results for compositional augmentation policy (other direction)

Results of our compositional augmentation policy in the direction of Upper Sorbian to German are shown in Table 6.

Formula	Weig. (λ)	Thre. (θ)	# of added sent.	BLEU	chrF2	TER
- (no aug.)	-	-	-	32.5 ± 1.1	60.2 ± 0.8	46.7 ± 1.0
Renorm.	-	0.45	1,303	33.4 ± 1.1	61.1 ± 0.8	46.1 ± 1.0
Renorm. LFW	0.2	0.30	147	33.8 ± 1.1	61.4 ± 0.8	45.6 ± 1.1
Avg. hapaxes	0.8	0.40	1,020	33.4 ± 1.1	60.8 ± 0.8	46.0 ± 1.0
Renorm. Stop.	0.9	0.30	22	33.7 ± 1.1	61.5 ± 0.8	46.0 ± 1.1
IDF-weig.	0.3	0.50	2,014	33.4 ± 1.1	61.0 ± 0.8	45.8 ± 1.0
Renorm. IDF-weig.	0.7	0.45	700	33.3 ± 1.1	61.1 ± 0.8	46.0 ± 1.0

Table 5 Same as Table 4, but in the other direction (Upper Sorbian to German)

Depth	Node No.	Parent Node No.	Total path (s)	Perplex.	Leaf node
0	0	-	-	18.3405	False
1	1	0	Avg. Hapax	18.0810	False
	2		IDF-weig.	18.0665	False
	3		Re. IDF-weig.	18.1784	False
	4		Re. LFW	17.8300	False
	5		Renorm.	17.9482	False
	6		Re. Stop.	18.0026	False
2	7	1	Avg. Hapax + Avg. Hapax	6.8171	True
	8		Avg. Hapax + IDF-weig.	6.8002	True
	9		Avg. Hapax + Re. IDF-weig.	6.8540	True
	10		Avg. Hapax + Re. LFW	6.7710	True
	11		Avg. Hapax + Renorm.	6.8088	True
	12		Avg. Hapax + Re. Stop.	6.9288	True
	13	2	IDF-weig. + Avg. Hapax	6.8379	True
	14		IDF-weig. + IDF-weig.	6.9974	True
	15		IDF-weig. + Re. IDF-weig.	6.8574	True
	16		IDF-weig. + Re. LFW	6.8816	True
	17		IDF-weig. + Renorm.	6.8618	True
	18		IDF-weig. + Re. Stop	6.8568	True
	19	3	Re. IDF-weig. + Avg. Hapax	6.8425	True
	20		Re. IDF-weig. + IDF-weig.	6.9477	True
	21		Re. IDF-weig. + Re. IDF-weig.	6.7993	True
	22		Re. IDF-weig. + Re. LFW	6.8758	True
	23		Re. IDF-weig. + Renorm.	6.9676	True
	24		Re. IDF-weig. + Re. Stop.	6.8312	True
	25	4	Re. LFW + IDF-weig.	6.8783	True
	26		Re. LFW + Re. IDF-weig.	6.8527	True
	27		Re. LFW + Re. Stop.	6.9639	True
	28	5	Renorm. + Renorm.	6.9077	True
	29	6	Re. Stop. + IDF-weig.	6.9482	True
	30		Re. Stop. + Re. IDF-weig.	6.8726	True
	31		Re. Stop. + Renorm.	6.8335	True
	32		Re. Stop. + Re. Stop.	6.9362	True

Table 6 Results of nodes in compositional policy (Upper Sorbian to German). Node with least perplexity is No. 10, generation path is formula for averaged hapaxes + renormalized LFW formula.