The SIGMORPHON 2020 Shared Task on Unsupervised Morphological Paradigm Completion

Katharina Kann[*]
University of Colorado Boulder
katharina.kann@colorado.edu

Arya D. McCarthy[*]
Johns Hopkins University
arya@jhu.edu

Garrett Nicolai
University of British Columbia
garrett.nicolai@ubc.ca

Mans Hulden
University of Colorado Boulder
mans.hulden@colorado.edu

Abstract

In this paper, we describe the findings of the SIGMORPHON 2020 shared task on unsupervised morphological paradigm completion (SIGMORPHON 2020 Task 2), a novel task in the field of inflectional morphology. Participants were asked to submit systems which take raw text and a list of lemmas as input, and output all inflected forms, i.e., the entire morphological paradigm, of each lemma. In order to simulate a realistic use case, we first released data for 5 development languages. However, systems were officially evaluated on 9 surprise languages, which were only revealed a few days before the submission deadline. We provided a modular baseline system, which is a pipeline of 4 components. 3 teams submitted a total of 7 systems, but, surprisingly, none of the submitted systems was able to improve over the baseline on average over all 9 test languages. Only on 3 languages did a submitted system obtain the best results. This shows that unsupervised morphological paradigm completion is still largely unsolved. We present an analysis here, so that this shared task will ground further research on the topic.

1 Introduction

In morphologically rich languages, words *inflect*: grammatical information like person, number, tense, and case are incorporated into the word itself, rather than expressed via function words. Not all languages mark the same properties: German nouns, for instance, have more inflected forms than their English counterparts.

When acquiring a language, humans usually learn to inflect words without explicit instruction. Thus, most native speakers are capable of generating inflected forms even of artificial lemmas (Berko, 1958). However, models that can generate paradigms without explicit morphological training have not yet been developed. We anticipate that such systems will be extremely useful, as they will open the possibility of rapid development of first-pass inflectional paradigms in a large set of languages. These can be utilized both *in se* for generation and as a starting point for elicitation (Sylak-Glassman et al., 2016), thus aiding the development of low-resource human language technologies (Christianson et al., 2018).

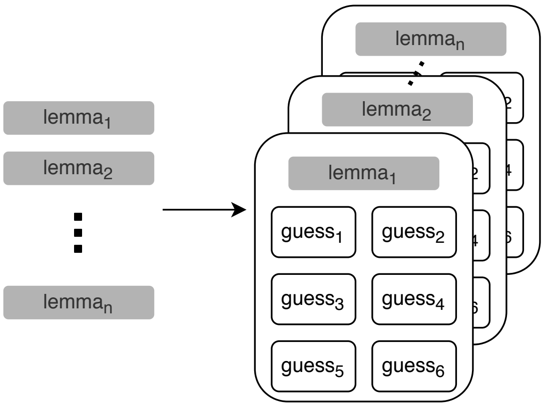

Figure 1: The task of *unsupervised morphological paradigm completion* (Jin et al., 2020) consists of generating complete inflectional paradigms for given lemmas, with the only additional available information being a corpus without annotations.

In this paper, we present the SIGMORPHON 2020 shared task on unsupervised morphological paradigm completion (SIGMORPHON 2020 Task 2). We asked participants to produce systems that can learn to inflect *in an unsupervised fashion*: given a small corpus (the Bible) together with a list of lemmas for each language, systems for the shared task should output all corresponding inflected forms. In their output, systems had to mark which forms expressed the same morphosyntactic features, e.g., demonstrate knowledge of the fact that *walks* is to *walk* as *listens* is to *listen*, despite not recognizing the morphological features explic-

[*] Equal contribution.

itly. We show a visualization of our shared task setup in Figure 1.

Unsupervised morphological paradigm completion requires solving multiple subproblems either explicitly or implicitly. First, a system needs to figure out which words in the corpus belong to the same paradigm. This can, for instance, be done via string similarity: *walks* is similar to *walk*, but less so to *listen*. Second, it needs to figure out the shape of the paradigm. This requires detecting which forms of different lemmas express the same morphosyntactic features, even if they are not constructed from their respective lemmas in the exact same way. Third, a system needs to generate all forms not attested in the provided corpus. Using the collected inflected forms as training data, this can be reduced to the supervised morphological inflection task (Cotterell et al., 2016).

This year's submitted systems can be split into two categories: those that built on the baseline (**Retrieval+X**) and those that did not (**Segment+Conquer**). The baseline system is set up as a pipeline which performs the following steps: edit tree retrieval, additional lemma retrieval, paradigm size discovery, and inflection generation (Jin et al., 2020). As it is highly modular, we provided two versions that employ different inflection models.[1] All systems built on the baseline substituted the morphological inflection component.

No system outperformed the baseline overall. However, two **Retrieval+X** models slightly improved over the baseline on three individual languages. We conclude that the task of unsupervised morphological paradigm completion is still an open challenge, and we hope that this shared task will inspire future research in this area.

2 Task and Evaluation

2.1 Unsupervised Morphological Paradigm Completion

Informal description. The task of unsupervised morphological paradigm completion mimics a setting where the only resources available in a language are a corpus and a short list of dictionary forms, i.e., lemmas. The latter could, for instance, be obtained via basic word-to-word translation. The goal is to generate all inflected forms of the given lemmas.

For an English example, assume the following lemma list to be given:

$$\begin{gathered} walk \\ listen \end{gathered}$$

With the help of raw text, systems should then produce an output like this:

$$\begin{array}{l} walk\ walk\ 1 \\ walk\ walks\ 2 \\ walk\ walked\ 3 \\ walk\ walking\ 4 \\ walk\ walked\ 5 \\ listen\ listens\ 2 \\ listen\ listened\ 5 \\ listen\ listened\ 3 \\ listen\ listening\ 4 \\ listen\ listen\ 1 \end{array} \tag{1}$$

The numbers serve as unique identifiers for paradigm slots: in above example, ”4” corresponds to the *present participle*. The inflections *walking* and *talking* therefore belong to the same paradigm slot. For the task, participants are not provided any knowledge of the grammatical content of the slots.

Formal definition. We denote the paradigm $\pi(\ell)$ of a lemma ℓ as

$$\pi(\ell) = \left\langle f(\ell, \vec{t_\gamma}) \right\rangle_{\gamma \in \Gamma(\ell)}, \tag{2}$$

with $f : \Sigma^* \times \mathcal{T} \rightarrow \Sigma^*$ being a function that maps a lemma and a vector of morphological features $\vec{t_\gamma} \in \mathcal{T}$ expressed by paradigm slot γ to the corresponding inflected form. $\Gamma(\ell)$ is the set of slots in lemma ℓ's paradigm.

We then formally describe the task of unsupervised morphological paradigm completion as follows. Given a corpus $\mathcal{D} = w_1, \ldots, w_{|\mathcal{D}|}$ together with a list $\mathcal{L} = \{\ell_j\}$ of $|\mathcal{L}|$ lemmas belonging to the same part of speech,[2] unsupervised morphological paradigm completion consists of generating the paradigms $\{\pi(\ell)\}$ of all lemmas $\ell \in \mathcal{L}$.

Remarks. It is impossible for unsupervised systems to predict the names of the features expressed by paradigm slots, an arbitrary decision made by human annotators. This is why, for the shared task,

[1] In this report, we use the words *baseline* and *baselines* interchangeably.

[2] This edition of the shared task was only concerned with verbs, though we are considering extending the task to other parts of speech in the future.

we asked systems to mark which forms belong to the same slot by numbering them, e.g., to predict that *walked* is the form for slot 3, while *listens* corresponds to slot 2.

2.2 Macro-averaged Best-Match Accuracy

The official evaluation metric was macro-averaged best-match accuracy (BMAcc; Jin et al., 2020).

In contrast to supervised morphological inflection (Cotterell et al., 2016), our task cannot be evaluated with word-level accuracy. For the former, one can compare the prediction for each lemma and morphological feature vector to the ground truth. However, for unsupervised paradigm completion, this requires a mapping from predicted slots to the gold standard's paradigm slots.

BMAcc, thus, first computes the word-level accuracy each predicted slot would obtain against each true slot. It then constructs a complete bipartite graph, with those accuracies as edge weights. This enables computing of the maximum-weight full matching with the algorithm of Karp (1980). BMAcc then corresponds to the sum of all accuracies for the best matching, divided by the maximum of the number of gold and predicted slots.

BMAcc penalizes systems for predicting a wrong number of paradigm slots. However, detecting the correct number of *identical* slots – something we encounter in some languages due to syncretism – is extremely challenging. Thus, we merge slots with identical forms for all lemmas in both the predictions and the ground truth before evaluating.

Example. Assume our gold standard is (1) (the complete, 5-slot English paradigms for the verbs *walk* and *listen*) and a system outputs the following, including an error in the fourth row:

$$\begin{aligned} &\textit{walk walks } 1\\ &\textit{walk walking } 2\\ &\textit{listen listens } 1\\ &\textit{listen listenen } 2 \end{aligned}$$

First, we merge slots 3 and 5 in the gold standard, since they are identical for both lemmas. Ignoring slot 5, we then compute the BMAcc as follows. Slot 1 yields an accuracy of 100% as compared to gold slot 2, and 0% otherwise. Similarly, slot 2 reaches an accuracy of 50% for gold slot 4, and 0% otherwise. Additionally, given the best mapping of those two slots, we obtain 0% accuracy for gold slots 1 and 3. Thus, the BMAcc is

$$\text{BMAcc} = \frac{1 + 0.5 + 0 + 0}{4} = 0.375 \quad (3)$$

3 Shared Task Data

3.1 Provided Resources

We provided data for 5 development and 9 test languages. The development languages were available for system development and hyperparameter tuning, while the test languages were released shortly before the shared task deadline. For the test languages, no ground truth data was available before system submission. This setup emulated a real-world scenario with the goal to create a system for languages about which we have no information.

For the raw text corpora, we leveraged the JHU Bible Corpus (McCarthy et al., 2020). This resource covers 1600 languages, which will enable future work to quickly produce systems for a large set of languages. Additionally, using the Bible allowed for a fair comparison of models across languages without potential confounds such as domain mismatch. 7 of the languages have only the New Testament available (approximately 8k sentences), and 7 have both the New and Old Testaments (approximately 31k sentences).

All morphological information was taken from UniMorph (Sylak-Glassman et al., 2015; Kirov et al., 2018), a resource which contains paradigms for more than 100 languages. However, this information was only accessible to the participants for the development languages. UniMorph paradigms were further used internally for evaluation on the test languages—this data was then released after the conclusion of the shared task.

3.2 Languages

During the development phase of the shared task, we released 5 languages to allow participants to investigate various design decisions: Maltese (MLT), Persian (FAS), Portuguese (POR), Russian (RUS), and Swedish (SWE). These languages are typologically and genetically varied, representing a number of verbal inflectional phenomena. Swedish and Portuguese are typical of Western European languages, and mostly exhibit fusional, suffixing verbal inflection. Russian, as an exemplar of Slavic languages, is still mostly suffixing, but does observe regular ablaut, and has considerable phonologically-conditioned allomorphy. Maltese is a Semitic language with a heavy Romance influence, and verbs

		MLT	FAS	POR	RUS	SWE
1	# Tokens in corpus	193257	227584	828861	727630	871707
2	# Types in corpus	16017	11877	31446	46202	25913
3	# Lemmas	20	100	100	100	100
4	# Lemmas in corpus	10	22	50	50	50
5	# Inflections	640	13600	7600	1600	1100
6	# Inflections in corpus	252	545	1037	306	276
7	Paradigm size	16	136	76	16	11
8	Paradigm size (merged)	15	132	59	16	11

Table 1: Dataset statistics: **development** languages. # Inflections=number of inflected forms in the gold file, token-based; # Inflections in corpus=number of inflections from the gold file which can be found in the corpus, token-based; Paradigm size=number of different morphological feature vectors in the dataset for the language; Paradigm size (merged)=paradigm size, but counting slots with all forms being identical only once.

		EUS	BUL	ENG	FIN	DEU	KAN	NAV	SPA	TUR
1	# Tokens in corpus	195459	801657	236465	685699	826119	193213	104631	251581	616418
2	# Types in corpus	18367	37048	7144	54635	22584	28561	18799	9755	59458
3	# Lemmas	20	100	100	100	100	20	100	100	100
4	# Lemmas in corpus	4	50	50	50	50	10	9	50	50
5	# Inflections	10446	5600	500	14100	2900	2612	3000	7000	12000
6	# Inflections in corpus	97	915	127	497	631	1040	54	630	986
7	Paradigm size	1659	56	5	141	29	85	30	70	120
8	Paradigm size (merged)	1658	54	5	141	20	59	30	70	120

Table 2: Dataset statistics: **test** languages. # Inflections=number of inflected forms in the gold file, token-based; # Inflections in corpus=number of inflections from the gold file which can be found in the corpus, token-based; Paradigm size=number of different morphological feature vectors in the dataset for the language; Paradigm size (merged)=paradigm size, but counting slots with all forms being identical only once.

combine templatic and suffixing inflection. Persian is mostly suffixing, but does allow for verbal inflectional prefixation, such as negation and marking subjunctive mood. Since the development languages were used for system tuning, their scores did not count towards the final ranking.

After a suitable period for system development and tuning, we released nine test languages: Basque (EUS), Bulgarian (BUL), English (ENG), Finnish (FIN), German (DEU), Kannada (KAN), Navajo (NAV), Spanish (SPA), and Turkish (TUR). Although these languages observe many features common to the development languages, such as fusional inflection, suffixation, and ablaut, they also cover inflectional categories absent in the development languages. Navajo, unlike any of the development languages, is strongly prefixing. Basque, Finnish, and Turkish are largely agglutinative, with long, complex affix chains that are difficult to identify through longest suffix matching. Furthermore, Finnish and Turkish feature vowel harmony and consonant gradation, which both require a method to identify allomorphs correctly to be able to merge different variants of the same paradigm slot.

3.3 Statistics

Statistics of the resources provided for all languages are shown in Table 1 for the development languages and in Table 2 for the test languages.

The token count (line 1) and, thus, the size of the provided Bible corpora, differs between 104,631 (Kannada) and 871,707 (Swedish). This number depends both on the typology of a language and on the completeness of the provided Bible translation. The number of types (line 2) is between 7,144 (English) and 59,458 (Turkish). It is strongly influenced by how morphologically rich a language is, i.e., how large the paradigms are, which is often approximated with the *type–token ratio*. The verbal paradigm size is listed in line 7: English has with a size of 5 the smallest paradigms, and, correspondingly, the lowest type count. Turkish, which has the highest number of types, in contrast, has large paradigms (120). The last line serves as an indicator of syncretism: subtracting line 8 from line 7 results in the number of paradigm slots that have been merged as a language evolved to use identical forms for different inflectional categories.

Lines 3 and 4 show the number of lemmas in the lemma lists for all languages, as well as the

Institution	Systems	Rank	Description Paper
KU-CST	KU-CST-1	7	Agirrezabal and Wedekind (2020)
KU-CST	KU-CST-2	6	Agirrezabal and Wedekind (2020)
IMS-CUBoulder	IMS-CUBoulder-1	5	Mager and Kann (2020)
IMS-CUBoulder	IMS-CUBoulder-2	1	Mager and Kann (2020)
NYU-CUBoulder	NYU-CUBoulder-1	4	Singer and Kann (2020)
NYU-CUBoulder	NYU-CUBoulder-2	2	Singer and Kann (2020)
NYU-CUBoulder	NYU-CUBoulder-3	3	Singer and Kann (2020)

Table 3: All submitted systems by institution, together with a reference to their description paper. The rank is relative to all other submitted systems and does not take the baselines into account.

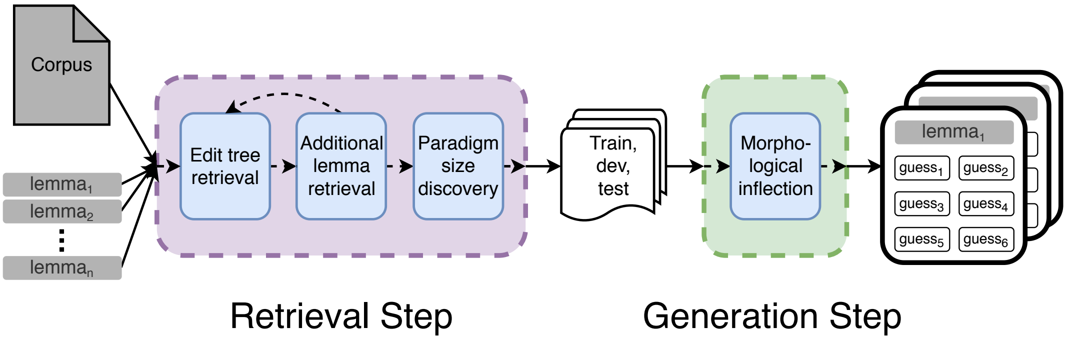

Figure 2: Our baseline system: the retrieval component bootstraps lemma–form–slot triplets, which are then used by the generation component to generate unobserved inflections in the paradigm of each input lemma.

number of lemmas which can be found in the corpus. For the majority of languages, 100 lemmas are provided, out of which 50 appear in the Bible. Exceptions are Maltese (20, 10), Persian (100, 22), Basque (20, 4), Kannada (20, 10), and Navajo (100, 9). These are due to limited UniMorph coverage.

In line 5, we list the number of total inflections, counting each one in the case of identical forms, i.e., this corresponds to the number of lines in our gold inflection file. English, due to its small verbal paradigm size, has only 500 inflections in our data. Conversely, Finnish has with 14,100 the largest number of inflections. Line 6 describes how many of the forms from line 5 appear in the corpus. As before, all forms are counted, even if they are identical. For all languages, a large majority of forms cannot be found in the corpus. This makes the task of unsupervised morphological paradigm completion with our provided data a challenging one.

4 Systems

In this section, we first review the baseline before describing the submitted systems. An additional overview of the submissions is shown in Table 3.

4.1 Baseline

We compared all submissions to the baseline system of Jin et al. (2020), graphically summarized in Figure 2. It is a pipeline system, which consists of 4 separate modules, which, in turn, can be grouped into two major components: *retrieval* and *generation*. The *retrieval component* discovers and returns inflected forms – and, less importantly, additional lemmas – from the provided Bible corpus. The *generation component* produces new inflected forms which cannot be found in the raw text.

The **retrieval component** performs three steps: First, it extracts the most common edit trees (Chrupała, 2008), i.e., it detects regularities with regards to word formation, based on the lemma list. If, for instance, both *walk* and *listen* are the lemmas provided and both *walked* and *listened* are encountered in the corpus, the system notes that appending *-ed* is a common transformation, which might correspond to an inflectional strategy.

Second, it retrieves new lemmas, with the goal to gather additional evidence for our collected edit trees. If, for instance, it has already identified the suffix *-ed* as an inflectional marker, finding both *pray* and *prayed* in the Bible is an indication that *pray* might be a lemma. New lemmas can then, in turn, be used to detect new regularities, e.g., in the case that *listen* and *listens* as well as *pray* and *prays* are attested in the corpus, but *walks* is not. Due to their complementary nature, components one and two can, as a unit, be applied iteratively to bootstrap a larger list of lemmas and transformations. For the baseline, we apply each of them only once.

Finally, the baseline's retrieval component predicts the paradigm size by analyzing which edit trees might be representing the same inflection. For instance, the suffixes *-d* and *-ed* both represent the past tense in English. The output of the retrieval component is a list of inflected forms with their lemmas, annotated with a paradigm slot number.

The **generation component** receives this output and prepares the data to train an inflectional generator. First, identified inflections are divided into a training and development split, and missing paradigm slots are identified. The generator is trained on the discovered inflections, and new forms are predicted for each missing slot.

We used two morphological inflection systems for the two variants of our baseline: the non-neural baseline from Cotterell et al. (2017) and the model proposed by Makarov and Clematide (2018). Both are highly suitable for the low-resource setting.

4.2 Submitted Systems: Retrieval+X

We now describe the first category of shared task submissions: Retrieval+X. Systems in this category leverage the retrieval component of the baseline, while substituting the morphological inflection component with a custom inflection system.

The **IMS–CUBoulder team** relied on LSTM (Hochreiter and Schmidhuber, 1997) sequence-to-sequence models for inflection. In IMS-CUB-1, the generation component is based on the architecture by Bahdanau et al. (2015), but with fewer parameters, as suggested by Kann and Schütze (2016). This model – as well as all other inflection components used for systems in this category – receives the sequence of the lemma's characters and the paradigm slot number as input and produces a sequence of output characters.

Their second system, IMS-CUB-2, uses an LSTM pointer-generator network (See et al., 2017) instead. This architecture has originally been proposed for low-resource morphological inflection by Sharma et al. (2018).

The **NYU–CUBoulder team** also substituted the baseline's generation component. Their morphological inflection models are ensembles of different combinations of transformer sequence-to-sequence models (Vaswani et al., 2017) and pointer-generator transformers, a model they introduced for the task.

NYU-CUB-1 is an ensemble of 6 pointer-generator transformers, while NYU-CUB-2 is an ensemble of 6 vanilla transformers. Their last system, NYU-CUB-3, is an ensemble of all 12 models.

4.3 Submitted Systems: Segment+Conquer

The **KU–CST team** did not modify the baseline directly, but, nevertheless, was heavily inspired by it. Their system first employs a character-segmentation algorithm to identify stem–suffix splits in both the provided lemma list and the corpus, thus identifying potential suffix-replacement rules. Next, k-means is used to cluster the extracted suffixes into allomorphic groups. These suffixes are then concatenated with the most frequent stems obtained from the lemma list, and scored by a language model, in order to arrive at plausible inflectional candidates. This approach is KU-CST-2.

However, KU-CST-2 often produces very small inflectional paradigms; unsurprisingly, given that the provided corpora are small as well, and, thus, any particular lemma is only inflected in limited ways – if at all. Therefore, KU-CST-1 expands the lemma list with a logistic-regression classifier that identifies novel verbs to be added.

5 Results and Analysis

5.1 Results on Development Languages

To encourage reproducibility, we first report the performance of all systems on the development languages in the upper part of Table 4. Although participants were not evaluated on these languages, the results provide insight and enable future researchers to benchmark their progress, while maintaining the held-out status of the test languages.

5.2 Official Shared Task Results

We show the official test results in the lower part of Table 4. Baseline-2 obtained the highest BMAcc on average, followed in order by Baseline-1, IMS-CUB-2, and NU-CUB-2. Overall, systems built on top of the baseline, i.e., systems from Retrieval+X, performed better than systems from Segment+Conquer: the best Segment+Conquer system only reached 4.66% BMAcc on average. This shows the effectiveness of the baseline. However, it also shows that we still have substantial room

	Baseline				KU-CST				IMS-CUB				NYU-CUB					
	1		2		1		2		1		2		1		2		3	
MLT	9.12	(17)	**20.00**	(17)	0.22	(254)	1.30	(2)	14.41	(17)	17.35	(17)	15.29	(17)	15.59	(17)	15.88	(17)
FAS	**6.67**	(31)	6.54	(31)	1.55	(11)	0.74	(2)	2.52	(31)	2.70	(31)	2.76	(31)	2.73	(31)	2.74	(31)
POR	**40.39**	(34)	39.56	(34)	1.09	(1104)	12.75	(70)	38.69	(34)	39.17	(34)	39.93	(34)	39.95	(34)	40.07	(34)
RUS	40.68	(19)	**41.68**	(19)	0.35	(387)	7.06	(10)	38.63	(19)	41.11	(19)	39.26	(19)	40.00	(19)	39.74	(19)
SWE	**45.07**	(15)	40.93	(15)	0.93	(588)	22.82	(17)	37.60	(15)	39.93	(15)	39.80	(15)	39.93	(15)	40.13	(15)
avg.	28.39		**29.74**		0.83		8.93		26.37		28.05		27.41		27.64		27.71	
EUS	0.06	(30)	0.06	(27)	0.02	(30)	0.01	(2)	0.04	(30)	0.06	(30)	0.05	(30)	0.05	(30)	**0.07**	(30)
BUL	28.30	(35)	31.69	(34)	2.99	(138)	4.15	(13)	27.22	(35)	**32.11**	(35)	27.69	(35)	28.94	(35)	27.89	(35)
ENG	65.60	(4)	**66.20**	(4)	3.53	(51)	17.29	(7)	47.80	(4)	61.00	(4)	50.20	(4)	52.80	(4)	51.20	(4)
FIN	5.33	(21)	**5.50**	(21)	0.39	(1169)	2.08	(108)	4.90	(21)	5.38	(21)	5.36	(21)	5.47	(21)	5.35	(21)
DEU	28.35	(9)	**29.00**	(9)	0.70	(425)	4.98	(40)	24.60	(9)	28.35	(9)	27.30	(9)	27.35	(9)	27.35	(9)
KAN	15.49	(172)	15.12	(172)	4.27	(44)	1.69	(1)	10.50	(172)	**15.65**	(172)	11.10	(172)	11.16	(172)	11.10	(172)
NAV	3.23	(3)	**3.27**	(3)	0.13	(38)	0.20	(2)	0.33	(3)	1.17	(3)	0.40	(3)	0.43	(3)	0.43	(3)
SPA	22.96	(29)	**23.67**	(29)	3.52	(225)	10.84	(40)	19.50	(29)	22.34	(29)	20.39	(29)	20.56	(29)	20.30	(29)
TUR	14.21	(104)	**15.53**	(104)	0.11	(1772)	0.71	(502)	13.54	(104)	14.73	(104)	14.88	(104)	15.39	(104)	15.13	(104)
avg.	20.39		**21.12**		1.74		4.66		16.49		20.09		17.49		18.02		17.65	

Table 4: BMAcc in percentages and the number of predicted paradigm slots after merging for all submitted systems and the baselines on all development (top) and test languages (bottom). Best scores are in bold.

for improvement on unsupervised morphological paradigm completion.

Looking at individual languages, Baseline-2 performed best for all languages except for EUS, where NYU-CUB-3 obtained the highest BMAcc, and BUL and KAN, where IMS-CUB-2 was best.

5.3 Analysis: Seen and Unseen Lemmas

We further look separately at the results for lemmas which appear in the corpus and those that do not. While seeing a lemma in context might help some systems, we additionally assume that inflections of attested lemmas are also more likely to appear in the corpus. Thus, we expect the performance for seen lemmas to be higher on average.

Examining the performance with respect to observed *inflected forms* might give cleaner results. However, we instead perform this analysis on a per-lemma basis, since the lemmas are part of a system's *input*, while the inflected forms are not.

Table 5 shows the performance of all systems for seen and unseen lemmas. Surprisingly, both versions of the baseline show similar BMAcc for both settings with a maximum difference of 0.12% on average. However, the baseline is the only system that performs equally well for unseen lemmas; IMS-CUB-1 observes the largest difference, with an absolute drop of 7.85% BMAcc when generating the paradigms of unseen lemmas. Investigating the cause for IMS-CUB-1's low BMAcc, we manually inspected the English output files, and found that, for unseen lemmas, many generations are nonsensical (e.g., *demoates* as an inflected form of *demodulate*). This does not happen in the case of seen lemmas. A similar effect has been found by Kann and Schütze (2018), who concluded that this might be caused by the LSTM sequence-to-sequence model not having seen similar character sequences during training. The fact that IMS-CUB-2, which uses another inflection model, performs better for unseen lemmas confirms this suspicion. Thus, additional training of the inflection component of IMS-CUB-1 on words from the corpus might improve generation. Conversely, the baseline – which benefits from inflection models specifically catered to low-resource settings – is better suited to inflecting unseen lemmas. Overall, we conclude that there is little evidence that the difficulty of the task increases for unseen lemmas. Rather, inflection systems need to compensate for the low contextual variety in their training data.

6 Where from and Where to?

6.1 Previous Work

Prior to this shared task, most research on unsupervised systems for morphology was concerned with developing approaches to segment words into morphemes, i.e., their smallest meaning-bearing units (Goldsmith, 2001; Creutz, 2003; Creutz and Lagus, 2007; Snyder and Barzilay, 2008; Goldwater et al., 2009; Kurimo et al., 2010; Kudo and Richardson, 2018). These methods were built around the observation that inflectional morphemes are very common across word types, and leveraged probabil-

	Baseline				KU-CST				IMS-CUB				NYU-CUB					
	1		2		1		2		1		2		1		2		3	
EUS	0.11	(30)	0.11	(19)	0.03	(30)	0.03	(2)	0.11	(28)	**0.19**	(30)	0.11	(30)	0.11	(30)	0.11	(30)
BUL	25.48	(35)	28.93	(34)	5.62	(138)	6.33	(13)	27.85	(35)	29.70	(34)	29.30	(35)	**29.78**	(35)	29.52	(35)
ENG	70.80	(4)	**71.20**	(4)	3.02	(51)	18.86	(7)	69.60	(4)	70.40	(4)	69.20	(4)	70.00	(4)	70.00	(4)
FIN	6.17	(21)	6.38	(21)	0.70	(1169)	3.60	(108)	6.11	(21)	**6.65**	(21)	6.55	(21)	6.58	(21)	6.57	(21)
DEU	26.70	(9)	27.00	(9)	1.14	(425)	8.75	(40)	27.40	(9)	27.30	(9)	27.50	(9)	**27.60**	(9)	27.40	(9)
KAN	16.35	(171)	15.61	(172)	6.61	(44)	1.69	(1)	13.99	(172)	**16.49**	(172)	14.63	(172)	14.68	(172)	14.63	(172)
NAV	**2.96**	(3)	**2.96**	(3)	1.46	(38)	2.22	(2)	**2.96**	(3)	**2.96**	(3)	**2.96**	(3)	**2.96**	(3)	**2.96**	(3)
SPA	20.97	(29)	**21.60**	(29)	4.43	(225)	16.37	(40)	20.40	(29)	21.14	(29)	21.17	(29)	21.09	(29)	21.14	(29)
TUR	14.68	(104)	16.38	(104)	0.23	(1772)	1.42	(502)	16.98	(104)	18.02	(104)	18.30	(104)	**18.70**	(104)	18.50	(104)
avg.	20.47		21.13		2.58		6.59		20.60		**21.43**		21.08		21.28		21.20	
EUS	0.06	(30)	0.06	(30)	0.03	(30)	0.00	(2)	0.03	(30)	0.04	(30)	0.05	(30)	0.05	(30)	**0.07**	(30)
BUL	31.11	(35)	34.44	(34)	0.83	(138)	2.04	(13)	26.59	(35)	**34.52**	(35)	26.07	(35)	28.11	(35)	26.26	(35)
ENG	60.40	(4)	**61.20**	(4)	4.12	(51)	15.71	(7)	26.00	(4)	51.60	(4)	31.20	(4)	35.60	(4)	32.40	(4)
FIN	4.52	(21)	**4.62**	(21)	0.12	(1169)	0.98	(108)	3.69	(21)	4.11	(21)	4.17	(21)	4.37	(21)	4.13	(21)
DEU	30.84	(9)	**32.63**	(9)	0.55	(425)	3.05	(40)	22.95	(9)	30.95	(9)	28.74	(9)	28.63	(9)	28.95	(9)
KAN	14.64	(172)	14.55	(172)	1.88	(24)	1.69	(1)	6.72	(172)	**14.72**	(172)	7.27	(172)	7.33	(172)	7.28	(172)
NAV	3.26	(3)	**3.30**	(3)	0.00	(38)	0.00	(2)	0.07	(3)	0.99	(3)	0.15	(3)	0.18	(3)	0.18	(3)
SPA	24.94	(29)	**25.74**	(29)	3.86	(225)	8.94	(40)	18.60	(29)	23.54	(29)	19.60	(29)	20.03	(29)	19.46	(29)
TUR	13.73	(104)	**14.70**	(104)	0.00	(1757)	0.00	(500)	10.12	(104)	11.47	(104)	11.48	(104)	12.08	(104)	11.77	(104)
avg.	20.39		**21.25**		1.27		3.60		12.75		19.10		14.30		15.15		14.50	

Table 5: BMAcc in percentages and the number of predicted paradigm slots after merging for all submitted systems and the baselines on all test languages; listed separately for lemmas which appear in the corpus (top) and lemmas which do not (bottom). Best scores are in bold.

ity estimates such as maximum likelihood (MLE) or maximum a posteriori (MAP) estimations to determine segmentation points, or minimum description length (MDL)-based approaches. However, they tended to make assumptions regarding how morphemes are combined, and worked best for purely concatenative morphology. Furthermore, these methods had no productive method of handling allomorphy—morphemic variance was simply treated as separate morphemes.

The task of unsupervised morphological paradigm completion concerns more than just segmentation: besides capturing how morphology is reflected in the word form, it also requires correctly clustering transformations into paradigm slots and, finally, generation of unobserved forms.

While Xu et al. (2018) did discover something similar to paradigms, those paradigms were a means to a segmentation end and the shape or size of the paradigms was not a subject of their research. Moon et al. (2009) similarly uses segmentation and clustering of affixes to group words into *conflation sets*, groups of morphologically related words, in an unsupervised way. Their work assumes prefixing and suffixing morphology. In a more task-driven line of research, Soricut and Och (2015) develop an approach to learn morphological transformation rules from observing how consistently word embeddings change between related word forms, with the goal of providing useful word embeddings for unseen words.

Our task further differs from traditional paradigm completion (e.g., Dreyer and Eisner, 2011; Ahlberg et al., 2015) in that *no* seed paradigms are observed. Thus, no information is being provided regarding the paradigm size, inflectional features, or relationships between lemmas and inflected forms. Other recent work (Nicolai and Yarowsky, 2019; Nicolai et al., 2020) learned fine-grained morphosyntactic tools from the Bible, though they leveraged supervision projected from higher-resource languages (Yarowsky et al., 2001; Täckström et al., 2013).

Past shared tasks. This task extends a tradition of SIGMORPHON shared tasks concentrating on inflectional morphology.

The first such task (Cotterell et al., 2016) encouraged participants to create inflectional tools in a typologically diverse group of 10 languages. The task was fully-supervised, requiring systems to learn inflectional morphology from a large annotated database. This task is similar to human learners needing to generate inflections of previously unencountered word forms, after having studied thousands of other types.

The second task (Cotterell et al., 2017) extended the first task from 10 to 52 languages and started to encourage the development of tools for the low-resource setting. While the first shared task approximated an adult learner with experience with thousands of word forms, low-resource inflection was closer to the language learner that has only studied a small number of inflections—however, it was closer to L2 learning than L1, as it still required training sets with lemma–inflection–slot triplets. The 2017 edition of the shared task also introduced a paradigm-completion subtask: participants were given partially observed paradigms and asked to generate missing forms, based on complete paradigms observed during training. This could be described as the supervised version of our unsupervised task, and notably did not require participants to identify inflected forms from raw text—a crucial step in L1 learning.

The third year of the shared task (Cotterell et al., 2018) saw a further extension to more than 100 languages and another step away from supervised learning, in the form of a contextual prediction task. This task stripped away inflectional annotations, requiring participants to generate an inflection solely utilizing a provided lemma and sentential cues. This task further imitated language learners, but extended beyond morphological learning to morphosyntactic incorporation. Furthermore, removing the requirement of an inflectional feature vector more closely approximated the generation step in our task. However, it was still supervised in that participants were provided with lemma–inflection pairs in context during training. We, in contrast, made no assumption of the existence of such pairs.

Finally, the fourth iteration of the task (McCarthy et al., 2019) again concentrated on less-supervised inflection. Cross-lingual training allowed low-resource inflectors to leverage information from high-resource languages, while a contextual analysis task flipped the previous year's contextual task on its head—tagging a sentence with inflectional information. This process is very similar to the retrieval portion of our task. We extended this effort to not only identify the paradigm slot of particular word, but to combine learned information from each class to extend and complete existing paradigms. Furthermore, we lifted the requirement of named inflectional features, more closely approximating the problem as approached by L1 language learners.

6.2 Future Shared Tasks

Future editions of the shared task could extend this year's Task 2 to a larger variety of languages or parts of speech. Another possible direction is to focus on derivational morphology instead of or in addition to inflectional morphology. We are also considering merging Task 2 with the traditional morphological inflection task: participants could then choose to work on the overall task or on either of the retrieval or generation subproblem.

Finally, we are looking into extending the shared task to use speech data as input. This is closer to how L1 learners acquire morphological knowledge, and, while this could make the task harder in some aspects, it could make it easier in others.

7 Conclusion

We presented the findings of the SIGMORPHON 2020 shared task on unsupervised morphological paradigm completion (SIGMORPHON 2020 Task 2), in which participants were asked to generate paradigms without explicit supervision.

Surprisingly, no team was able to outperform the provided baseline, a pipeline system, on average over all test languages. Even though 2 submitted systems were better on 3 individual languages, this highlights that the task is still an open challenge for the NLP community. We argue that it is an important one: systems obtaining high performance will be able to aid the development of human language technologies for low-resource languages.

All teams that participated in the shared task devised modular approaches. Thus, it will be easy to include improved components in the future as, for instance, systems for morphological inflection improve. We released all data, the baseline, the evaluation script, and the system outputs in the official repository,[3] in the hope that this shared task will lay the foundation for future research on unsupervised morphological paradigm completion.

Acknowledgments

First and foremost, we would like to thank all of our shared task participants. We further thank the passionate morphologists who joined for lunch in Florence's *mercato centrale* on the last day of ACL 2019 to plan the 2020 shared task, as well as the SIGMORPHON Exec, who made this shared task possible.

[3] https://github.com/sigmorphon/2020/tree/master/task2

References

Manex Agirrezabal and Jürgen Wedekind. 2020. KU-CST at the SIGMORPHON 2020 task 2 on unsupervised morphological paradigm completion. In *Proceedings of the 17th Workshop on Computational Research in Phonetics, Phonology, and Morphology*. Association for Computational Linguistics.

Malin Ahlberg, Markus Forsberg, and Mans Hulden. 2015. Paradigm classification in supervised learning of morphology. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1024–1029, Denver, Colorado. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Jean Berko. 1958. The child's learning of english morphology. *Word*, 14(2-3):150–177.

Caitlin Christianson, Jason Duncan, and Boyan Onyshkevych. 2018. Overview of the DARPA LORELEI program. *Machine Translation*, 32(1):3–9.

Grzegorz Chrupała. 2008. *Towards a machine-learning architecture for lexical functional grammar parsing*. Ph.D. thesis, Dublin City University.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Arya D. McCarthy, Katharina Kann, Sebastian Mielke, Garrett Nicolai, Miikka Silfverberg, David Yarowsky, Jason Eisner, and Mans Hulden. 2018. The CoNLL–SIGMORPHON 2018 shared task: Universal morphological reinflection. In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 1–27, Brussels. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2017. CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages. In *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pages 1–30, Vancouver. Association for Computational Linguistics.

Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, and Mans Hulden. 2016. The SIGMORPHON 2016 shared Task—Morphological reinflection. In *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 10–22, Berlin, Germany. Association for Computational Linguistics.

Mathias Creutz. 2003. Unsupervised segmentation of words using prior distributions of morph length and frequency. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 280–287, Sapporo, Japan. Association for Computational Linguistics.

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Trans. Speech Lang. Process.*, 4(1).

Markus Dreyer and Jason Eisner. 2011. Discovering morphological paradigms from plain text using a Dirichlet process mixture model. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 616–627, Edinburgh, Scotland, UK. Association for Computational Linguistics.

John Goldsmith. 2001. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–198.

Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2009. A Bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, 112(1):21 – 54.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Huiming Jin, Liwei Cai, Yihui Peng, Chen Xia, Arya D. McCarthy, and Katharina Kann. 2020. Unsupervised morphological paradigm completion. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Katharina Kann and Hinrich Schütze. 2016. Single-model encoder-decoder with explicit morphological representation for reinflection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 555–560, Berlin, Germany. Association for Computational Linguistics.

Katharina Kann and Hinrich Schütze. 2018. Neural transductive learning and beyond: Morphological generation in the minimal-resource setting. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3254–3264, Brussels, Belgium. Association for Computational Linguistics.

Richard M. Karp. 1980. An algorithm to solve the $m \times n$ assignment problem in expected time $O(mn \log n)$. *Networks*, 10(2):143–152.

Christo Kirov, Ryan Cotterell, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sebastian Mielke, Arya McCarthy, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden. 2018. UniMorph 2.0: Universal morphology. In *Proceedings of the Eleventh*

International Conference on Language Resources and Evaluation (LREC-2018), Miyazaki, Japan. European Languages Resources Association (ELRA).

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Mikko Kurimo, Sami Virpioja, Ville Turunen, and Krista Lagus. 2010. Morpho challenge 2005-2010: Evaluations and results. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 87–95, Uppsala, Sweden. Association for Computational Linguistics.

Manuel Mager and Katharina Kann. 2020. The IMS–CUBoulder system for the SIGMORPHON 2020 shared task on unsupervised morphological paradigm completion. In *Proceedings of the 17th Workshop on Computational Research in Phonetics, Phonology, and Morphology*. Association for Computational Linguistics.

Peter Makarov and Simon Clematide. 2018. Imitation learning for neural morphological string transduction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2877–2882, Brussels, Belgium. Association for Computational Linguistics.

Arya D. McCarthy, Ekaterina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Christo Kirov, Miikka Silfverberg, Sebastian J. Mielke, Jeffrey Heinz, Ryan Cotterell, and Mans Hulden. 2019. The SIGMORPHON 2019 shared task: Morphological analysis in context and cross-lingual transfer for inflection. In *Proceedings of the 16th Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 229–244, Florence, Italy. Association for Computational Linguistics.

Arya D. McCarthy, Rachel Wicks, Dylan Lewis, Aaron Mueller, Winston Wu, Oliver Adams, Garrett Nicolai, Matt Post, and David Yarowsky. 2020. The Johns Hopkins University Bible Corpus: 1600+ tongues for typological exploration. In *Proceedings of the Twelfth International Conference on Language Resources and Evaluation (LREC 2020)*. European Language Resources Association (ELRA).

Taesun Moon, Katrin Erk, and Jason Baldridge. 2009. Unsupervised morphological segmentation and clustering with document boundaries. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 668–677, Singapore. Association for Computational Linguistics.

Garrett Nicolai, Dylan Lewis, Arya D. McCarthy, Aaron Mueller, Winston Wu, and David Yarowsky. 2020. Fine-grained morphosyntactic analysis and generation tools for more than one thousand languages. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3963–3972, Marseille, France. European Language Resources Association.

Garrett Nicolai and David Yarowsky. 2019. Learning morphosyntactic analyzers from the Bible via iterative annotation projection across 26 languages. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1765–1774, Florence, Italy. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Abhishek Sharma, Ganesh Katrapati, and Dipti Misra Sharma. 2018. IIT(BHU)–IIITH at CoNLL–SIGMORPHON 2018 shared task on universal morphological reinflection. In *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pages 105–111, Brussels. Association for Computational Linguistics.

Assaf Singer and Katharina Kann. 2020. The NYU–CUBoulder systems for SIGMORPHON 2020 Task 0 and Task 2. In *Proceedings of the 17th Workshop on Computational Research in Phonetics, Phonology, and Morphology*. Association for Computational Linguistics.

Benjamin Snyder and Regina Barzilay. 2008. Unsupervised multilingual learning for morphological segmentation. In *Proceedings of ACL-08: HLT*, pages 737–745, Columbus, Ohio. Association for Computational Linguistics.

Radu Soricut and Franz Och. 2015. Unsupervised morphology induction using word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1627–1637, Denver, Colorado. Association for Computational Linguistics.

John Sylak-Glassman, Christo Kirov, and David Yarowsky. 2016. Remote elicitation of inflectional paradigms to seed morphological analysis in low-resource languages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 3116–3120, Portorož, Slovenia. European Language Resources Association (ELRA).

John Sylak-Glassman, Christo Kirov, David Yarowsky, and Roger Que. 2015. A language-independent feature schema for inflectional morphology. In *Pro-*

ceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 674–680, Beijing, China. Association for Computational Linguistics.

Oscar Täckström, Dipanjan Das, Slav Petrov, Ryan McDonald, and Joakim Nivre. 2013. Token and type constraints for cross-lingual part-of-speech tagging. *Transactions of the Association for Computational Linguistics*, 1:1–12.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Hongzhi Xu, Mitchell Marcus, Charles Yang, and Lyle Ungar. 2018. Unsupervised morphology learning with statistical paradigms. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 44–54, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the First International Conference on Human Language Technology Research*.