Paradigm Clustering with Weighted Edit Distance

Andrew Gerlach, Adam Wiemerslage and **Katharina Kann**
University of Colorado Boulder
first.last@colorado.edu

Abstract

This paper describes our system for the SIGMORPHON 2021 Shared Task on Unsupervised Morphological Paradigm Clustering, which asks participants to group inflected forms together according their underlying lemma without the aid of annotated training data. We employ agglomerative clustering to group word forms together using a metric that combines an orthographic distance and a semantic distance from word embeddings. We experiment with two variations of an edit distance-based model for quantifying orthographic distance, but, due to time constraints, our systems do not outperform the baseline. However, we also show that, with more time, our results improve strongly.

1 Introduction

Most of the world's languages express grammatical properties, such as tense or case, via small changes to a word's surface form. This process is called *morphological inflection*, and the canonical form of a word is known as its *lemma*. A search of the WALS database of linguistic typology shows that 80% of the database's languages mark verb tense and 65% mark grammatical case through morphology (Dryer and Haspelmath, 2013).

The English lemma *do*, for instance, has an inflected form *did* that expresses past tense. Though English verbs inflect to express tense, there are generally only 4 to 5 surface variations for a given English lemma. In contrast, a Russian verb can have up to 30 morphological inflections per lemma, and other languages – such as Basque – have hundreds of forms per lemma, cf. Table 1.

Basque Lemma: *egin*		
begi	begiate	begidate
begie	begiete	begigu
begigute	begik	begin
beginate	begio	begiote
begit	begite	begitza
...	...	...
zenegizkigukeen	zenegizkigukete	zenegizkiguketen
zenegizkigun	zenegizkigute	zenegizkiguten
zenegizkio	zenegizkioke	zenegizkiokeen
zenegizkiokete	zenegizkioketen	zenegizkion
zenegizkiote	zenegizkioten	zenegizkit

Table 1: The paradigm of the Basque verb *egin* consists of 674 inflected forms. In contrast, the paradigm of the English verb *do* only consists of 5 inflected forms: *do*, *does*, *doing*, *did*, and *done*.

Inflected forms are systematically related to each other: in English, most noun plurals are obtained from the lemma by adding *-s* or *-es* to the end of the noun, e.g., *list*/*lists* or *kiss*/*kisses*. However, irregular plurals also exist, such as *ox*/*oxen* or *mouse*/*mice*. Although irregular forms are less frequent, they cause challenges for the automatic generation or analysis of the surface forms of English plural nouns.

In this work, we address the SIGMORPHON 2021 Shared Task on Unsupervised Morphological Paradigm Clustering ("Task 2") (Wiemerslage et al., 2021). The goal of this shared task is to group words encountered in naturally occurring text into morphological paradigms. Unsupervised paradigm clustering can be helpful for state-of-the-art natural language processing (NLP) systems, which typically require large amounts of training data. The ability to group words together into paradigms is a useful first step for training a system to induce full paradigms from a limited number of examples, a task known as (supervised) morphological paradigm completion. Building paradigms can help an NLP system

Proceedings of the Seventeenth SIGMORPHON Workshop on Computational Research

August 5, 2021. ©2

in Phonetics, Phonology, and Morphology,pages 107–114

to induce representations for rare words or to generate words that have not been observed in a given corpus. Lastly, unsupervised systems have the advantage of not needing annotated data, which can be costly in terms of time and money, or, in the case of extinct or endangered languages, entirely impossible.

Since 2016, the Association for Computational Linguistics' Special Interest Group on Computational Morphology and Phonology (SIGMORPHON) has created shared tasks to help spur the development of state-of-the-art systems to explicitly handle morphological processes in a language. These tasks have involved morphological inflection (Cotterell et al., 2016), lemmatization (McCarthy et al., 2019), as well as other, related tasks. SIGMORPHON has increased the level of difficulty of the shared tasks, largely along two dimensions. The first dimension is the amount of data available for models to learn, reflecting the difficulties of analyzing low-resource languages. The second dimension is the amount of structure provided in the input data. Initially, SIGMORPHON shared tasks provided predefined tables of lemmas, morphological tags, and inflected forms. For the SIGMORPHON 2021 Shared Task on Unsupervised Morphological Paradigm Clustering, only raw text is provided as input.

We propose a system that combines orthographic and semantic similarity measures to cluster surface forms found in raw text. We experiment with a character-level language model for weighing substring differences between words. Due to time constraints we are only able to cluster over a subset of each languages' vocabulary. Despite of this, our system's performance is comparable to the baseline.

2 Related Work

Unsupervised morphology has attracted a great deal of interest historically, including a large body of work focused on segmentation (Xu et al., 2018; Creutz and Lagus, 2007; Poon et al., 2009; Narasimhan et al., 2015). Recently, the task of unsupervised morphological paradigm completion has been proposed (Kann et al., 2020; Jin et al., 2020; Erdmann et al., 2020), wherein the goal is to induce full paradigms from raw text corpora.

In this year's SIGMORPHON shared task, we are asked to only address part of the unsupervised paradigm completion task: paradigm clustering. Intuitively, the task of segmentation is related to paradigm clustering, but the outputs are different. Goldsmith (2001) produces morphological signatures, which are similar to approximate paradigms, based on an algorithm that uses minimum description length. However, this type of algorithm relies heavily on purely orthographic features of the vocabulary. Schone and Jurafsky (2001) hypothesize that approximating semantic information can help differentiate between hypothesized morphemes, revealing those that are productive. They propose an algorithm that combines orthography, semantics, and syntactic distributions to induce morphological relationships. They used semantic relatedness, quantified by latent semantic analysis, combined with the frequencies of affixes and syntactic context (Schone and Jurafsky, 2000).

More recently, Soricut and Och (2015) have used SkipGram word embeddings (Mikolov et al., 2013) to find meaningful morphemes based on analogies: regularities exhibited by embedding spaces allow for inferences of certain types (e.g., *king* is to *man* what *queen* is to *woman*). Hypothesizing that these regularities also hold for morphological relations, they represent morphemes by vector differences between semantically similar forms, e.g., the vector for the suffix $\overrightarrow{s}$ may be represented by the difference between $\overrightarrow{cats}$ and $\overrightarrow{cat}$.

Drawing upon these intuitions, we follow Rosa and Zabokrtský (2019), which combines semantic distance using fastText embeddings (Bojanowski et al., 2017) with an orthographic distance between word pairs. Words are then clustered into paradigms using agglomerative clustering.

3 Task Description

Given a raw text corpus, the task is to sort words into clusters that correspond to paradigms. More formally, for the vocabulary Σ of all types attested in the corpus and the set of morphological paradigms Π for which at least one word is in Σ, the goal is to output clusters corresponding to $\pi_k \bigcap \Sigma$ for all $\pi_k \in \Pi$.

Data As the raw text data for this task, JHU Bible corpora (McCarthy et al., 2020b) are provided by the organizers. This is the only data that systems can use. The organizers further provide development and test sets consisting of gold clusters for a subset of words in the Bible corpora. Each cluster is a list of words representing $\pi_k \bigcap \Sigma$ for $\pi_k \in \Pi_{dev}$ or $\pi_k \in \Pi_{test}$, respectively, and $\Pi_{dev}, \Pi_{test} \subsetneq \Pi$.

The partial morphological paradigms in Π_{dev} and Π_{test} are taken from the UniMorph database (McCarthy et al., 2020a). Development sets are only available for the development languages, while test sets are only provided for the test languages. All test sets are hidden from the participants until the conclusion of the shared task.

Languages The development languages featured in the shared task are Maltese, Persian, Portuguese, Russian, and Swedish. The test languages are Basque, Bulgarian, English, Finnish, German, Kannada, Navajo, Spanish, and Turkish.

4 System Descriptions

We submit two systems based on Rosa and Zabokrtský (2019). The first, referred to below as *JW-based clustering*, follows their work very closely. The second, *LM-based clustering*, contains the same main components, but approximates orthographic distances with the help of a language model.

4.1 JW-based Clustering

We describe the system of Rosa and Zabokrtský (2019) in more detail here. This system clusters over words whose distance is computed as a combination of orthographic and semantic distances.

Orthographic Distance The orthographic distance of two words is computed as their Jaro-Winkler (JW) edit distance (Winkler, 1990). JW distance differs from the more common Levenshtein distance (Levenshtein, 1966) in that JW distance gives more importance to the beginnings of strings than to their ends, which is where characters belonging to the stem are likely to be in suffixing languages.

The JW distance is averaged with the JW distance of a *simplified variant* of the string. The simplified variant is a string that has been lower cased, transliterated to ASCII, and had the non-initial vowels deleted. This is done to soften the impact of characters that are likely to correspond with affixes. Crucially, we believe that this biases the system towards languages that express inflection via suffixation.

Semantic Distance We represent words in the corpus by fastText embeddings, similar to Erdmann and Habash (2018), who cluster fastText embeddings for the same task in various Arabic dialects. We expect fastText embeddings to provide better representations than, e.g., Word2Vec (Mikolov et al., 2013), due to the limited size of the Bible corpora. Unfortunately, using fastText may also inadvertently result in higher similarity between words belonging to different lemmas that contain overlapping subwords corresponding to affixes.

Overall Distance We compute a pairwise distance matrix for all words in the corpus. The distance between two words w_1 and w_2 is computed as:

$$d(w_1, w_2) = 1 - \delta(w_1, w_2) \cdot \frac{\cos(\hat{w}_1, \hat{w}_2) + 1}{2}, \quad (1)$$

where $\hat{w}_1$ and $\hat{w}_2$ are the embeddings of w_1 and w_2, cos is the cosine distance, and δ is the JW edit distance. The cosine distance is mapped to $[0, 1]$ to avoid negative distances.

Finally, agglomerative clustering is performed by first assigning each word form to a unique cluster. At each step, the two clusters

with the lowest average distance are merged together. The merging continues while the distance between clusters stays below a threshold. We tune this hyperparameter on the development set, and our final threshold is 0.3.

4.2 LM-based Clustering

The JW-based clustering described above relies on heuristics to obtain a good measure of orthographic similarity. These heuristics help to quantify orthographic similarity between two words by relying more on the shared characters in the stem than in the affix: The plural past participles *gravados* and *louvados* in Portuguese have longer substrings in common than the substrings by which they differ. This is due to the affix *-ados*, which indicates that the two words express the same inflectional information, even though their lemmas are different. Similarly, the Portuguese verbs *abafa* and *abafávamos* differ in many characters, though they belong to the same paradigm, as can be observed by the shared stem *abaf*.

However, not all languages express inflection exclusively via suffixation, nor via concatenation. We thus experiment with removing the edit distance heuristics and, instead, utilizing probabilities from a character-level language model (LM) to distinguish between stems and affixes. In doing so, we hope to achieve better results for templatic languages, such as Maltese. We hypothesize that the LM will have a higher confidence for characters that are part of an affix than for those that are part of the stem. We then draw upon this hypothesis and weigh edit operations between two strings based on these confidences.

LM-weighted Edit Distance Similar to the intuition behind Silfverberg and Hulden (2018), we train a character-level LM on the entire vocabulary for each Bible corpus. Unlike their work, we do not have inflectional tags for each word. Despite this, we hypothesize that the highly regular and frequent nature of inflectional affixes will lead to higher likelihoods for characters that occur in affixes than for those in stems. We train a two-layer LSTM (Hochreiter and Schmidhuber, 1997) with an embedding size of 128 and a hidden layer size of 128. We train the model until the training loss stops decreasing, for up to 100 epochs, using Adam (Kingma and Ba, 2014) with a learning rate of 0.001 and a batch size of 16.

When calculating the edit distance between two words, the insertion, deletion, or substitution costs are computed as a function of the LM probabilities. We expect this to give more weights to differences in the stem than to those in other parts of the word. Each character is then associated with a cost given by

$$\text{cost}(w_i) = 1 - \frac{p(w_i)}{\sum_{j \in |w|} p(w_j)}, \qquad (2)$$

where $p(w_i)$ is the probability of the ith character in word w as given by the LM. We then compute the cost of an insertion or deletion as the cost of the character being inserted or deleted. The cost of a substitution is the average of the costs of the two involved characters. The sum over these operations is the weighted edit distance between two words, $\epsilon(w_1, w_2)$. Finally, we compute pairwise distances using Equation 1, replacing $\delta(w_1, w_2)$ with

$$\frac{\epsilon(w_1, w_2)}{max(|w_1|, |w_2|)}.$$

Forward vs. Backward LM We hypothesize that the direction in which the LM is trained affects the probabilities for affixes. Intuitively, an LM is likely to assign higher confidence to characters at the beginning of a word than at the end. Thus, an LM trained on data in the forward direction (LM-F) should be more likely to assign higher probabilities to characters at the beginning of a word, such as prefixes, while a model trained on reversed words (LM-B) should assign higher probabilities to suffixes. In practice, LM-B outperforms LM-F on all development languages, cf. Table 2. Because of that, we employ LM-B to weigh edit operations for all test languages.[1]

[1]This might be caused by none of the development languages being prefixing. However, in order to make a more informed choice, a method to automatically distinguish between prefixing and suffixing languages from raw text alone would be necessary.

Lang	Baseline			LMC-B			LMC-F			JWC		
	prec.	rec.	F1	prec.	rec.	f1	prec.	rec.	F1	prec.	rec.	F1
Maltese	0.250	0.348	0.291	0.465	0.229	0.307	0.411	0.202	0.272	0.489	0.241	**0.323**
Persian	0.265	0.348	0.300	0.321	0.307	0.314	0.494	0.197	0.282	0.579	0.231	**0.330**
Portuguese	0.218	0.794	0.341	0.771	0.248	**0.376**	0.494	0.159	0.241	0.742	0.239	0.362
Russian	0.234	0.807	0.363	0.802	0.282	**0.417**	0.726	0.255	0.378	0.792	0.278	0.412
Swedish	0.303	0.776	0.436	0.818	0.378	0.517	0.695	0.321	0.439	0.838	0.388	**0.530**
Average	0.254	0.615	0.346	0.635	0.289	0.386	0.482	0.186	0.268	0.688	0.275	**0.391**

Table 2: Precision, recall, and F1 for all development languages. LMC-R is the LM-clustering system for language models trained from left-to-right (reverse). LMC-F are trained from left-to-right, and JWC is the JW-clustering system. The highest F1 for each language is in bold.

Lang	Baseline			LMC			JWC		
	prec.	rec.	F1	prec.	rec.	F1	prec.	rec.	F1
English	0.388	0.767	**0.515**	0.565	0.245	0.3420	0.663	0.288	0.402
Navajo	0.230	0.598	**0.333**	0.686	0.112	0.1928	0.657	0.108	0.185
Spanish	0.266	0.722	**0.388**	0.664	0.183	0.2869	0.699	0.193	0.302
Finnish	0.179	0.767	0.290	0.694	0.227	**0.342**	0.674	0.220	0.332
Bulgarian	0.265	0.730	0.390	0.745	0.312	**0.440**	0.717	0.300	0.423
Basque	0.186	0.254	0.215	0.471	0.254	**0.330**	0.353	0.191	0.247
Kannada	0.172	0.385	0.238	0.570	0.169	0.261	0.625	0.185	**0.286**
German	0.254	0.776	0.382	0.7626	0.310	0.441	0.787	0.319	**0.454**
Turkish	0.156	0.658	0.252	0.6574	0.212	**0.320**	0.641	0.206	0.312
Average	0.233	0.629	**0.334**	0.646	0.225	0.328	0.646	0.223	0.327

Table 3: Precision, recall, and F1 for all test languages. LMC is the LM-clustering system, JWC is the JW-clustering system. The highest F1 for each language is in bold.

5 Results and Discussion

The official scores obtained by our systems as well as the baseline are shown in Table 3.

Both of our systems perform minimally worse than the baseline if we consider F1 averaged over languages (0.334 vs. 0.328 and 0.327). However, we believe this to be largely due to our submissions only generating clusters for a subset of the full vocabularies: due to time constraints, we only consider words that appear at least 5 times in the corpus. No other words are included in the predicted clusters. The large gap between precision and recall reflects this constraint: our submissions have a high average precision (0.646 for both systems), indicating that the limited set of words we consider are being clustered more accurately than the F1 scores would suggest. The low recall scores (0.225 and 0.223) are likely at least partially caused by the missing words in our predictions.[2]

Conversely, the baseline system has a high recall (0.629) and a low precision (0.233). This is likely due to it simply clustering words with shared substrings, such that a given word is likely to appear in many predicted clusters.

Interestingly, both of our submissions have the same average precision on the test set, despite varying across languages. Notably, the LM-based clustering system strongly outperforms the JW-based system on Basque with respect to precision. However, the JW-based system outperforms the LM-based one by a large margin on English. One hypothesis for the difference in results is that agglutinating inflection in Basque causes very long affixes, which our LM-based system should downweigh in its measurement of orthographic similarity. Basque is also not a strictly suffixing language, which we expect the JW-based model to be biased towards. On the other hand, English has relatively little inflectional morphology, and is strictly suffixing (in terms of inflection). The assumptions behind the JW-based system are more ideal for a language like English. The JW system performs best on Maltese, which suggests that the heuristics of that system are sufficient for a templatic language, compared to the LM-based system.

[2]We confirm this hypothesis with additional experiments after the shared task's completion. Those results can be found in the appendix.

6 Conclusion

We present two systems for the SIGMORPHON 2021 Shared Task on Unsupervised Morphological Paradigm Clustering. Both of our systems perform slighly worse than the official baseline. However, we also show that this is due to our official submissions only making predictions for a subset of the corpus' vocabulary, due to time constraints and that at least one of our systems improves strongly if the time constraints are removed.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

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 2016 Meeting of SIGMORPHON*, Berlin, Germany. Association for Computational Linguistics.

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. 4(1).

Matthew S. Dryer and Martin Haspelmath, editors. 2013. *WALS Online*. Max Planck Institute for Evolutionary Anthropology, Leipzig.

Alexander Erdmann, Micha Elsner, Shijie Wu, Ryan Cotterell, and Nizar Habash. 2020. The paradigm discovery problem. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7778–7790, Online. Association for Computational Linguistics.

Alexander Erdmann and Nizar Habash. 2018. Complementary strategies for low resourced morphological modeling. In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 54–65, Brussels, Belgium. Association for Computational Linguistics.

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

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 McCarthy, and Katharina Kann. 2020. Unsupervised morphological paradigm completion. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6696–6707, Online. Association for Computational Linguistics.

Katharina Kann, Arya D. McCarthy, Garrett Nicolai, and Mans Hulden. 2020. The SIGMORPHON 2020 shared task on unsupervised morphological paradigm completion. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 51–62, Online. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Vladimir Iosifovich Levenshtein. 1966. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10(8):707–710. Doklady Akademii Nauk SSSR, V163 No4 845-848 1965.

Arya D. McCarthy, Christo Kirov, Matteo Grella, Amrit Nidhi, Patrick Xia, Kyle Gorman, Ekaterina Vylomova, Sabrina J. Mielke, Garrett Nicolai, Miikka Silfverberg, Timofey Arkhangelskiy, Nataly Krizhanovsky, Andrew Krizhanovsky, Elena Klyachko, Alexey Sorokin, John Mansfield, Valts Ernštreits, Yuval Pinter, Cassandra L. Jacobs, Ryan Cotterell, Mans Hulden, and David Yarowsky. 2020a. UniMorph 3.0: Universal Morphology. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3922–3931, Marseille, France. European Language Resources Association.

Arya D. McCarthy, Ekaterina Vylomova, Shijie Wu, Chaitanya Malaviya, Lawrence Wolf-Sonkin, Garrett Nicolai, Christo Kirov, Miikka Silfverberg, Sabrina 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. 2020b. The Johns Hopkins University Bible corpus: 1600+ tongues for typological exploration. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2884–2892, Marseille, France. European Language Resources Association.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

Karthik Narasimhan, Regina Barzilay, and Tommi Jaakkola. 2015. An unsupervised method for uncovering morphological chains. *Transactions of the Association for Computational Linguistics*, 3:157–167.

Hoifung Poon, Colin Cherry, and Kristina Toutanova. 2009. Unsupervised morphological segmentation with log-linear models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 209–217, Boulder, Colorado. Association for Computational Linguistics.

Rudolf Rosa and Zdenek Zabokrtský. 2019. Unsupervised lemmatization as embeddings-based word clustering. *CoRR*, abs/1908.08528.

Patrick Schone and Daniel Jurafsky. 2000. Knowledge-free induction of morphology using latent semantic analysis. In *Fourth Conference on Computational Natural Language Learning and the Second Learning Language in Logic Workshop*.

Patrick Schone and Daniel Jurafsky. 2001. Knowledge-free induction of inflectional morphologies. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Miikka Silfverberg and Mans Hulden. 2018. An encoder-decoder approach to the paradigm cell filling problem. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2883–2889, Brussels, Belgium. 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.

Adam Wiemerslage, Arya McCarthy, Alexander Erdmann, Garrett Nicolai, Manex Agirrezabal, Miikka Silfverberg, Mans Hulden, and Katharina Kann. 2021. The SIGMORPHON 2021 shared task on unsupervised morphological paradigm clustering. In *Proceedings of the 18th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*. Association for Computational Linguistics.

William E. Winkler. 1990. String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. In *Proceedings of the Section on Survey Research*, pages 354–359.

Ruochen Xu, Yiming Yang, Naoki Otani, and Yuexin Wu. 2018. Unsupervised cross-lingual transfer of word embedding spaces. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2465–2474, Brussels, Belgium. Association for Computational Linguistics.

7 Appendix

Here we present new results which include the entire data set for selected languages. We see an improvement in F1 for each language. This due to the increased recall scores from the paradigms being more complete. Precision scores decrease across the board. This may be due to the languages being sensitive to the threshold value.

Lang	**Subset**			**Full**		
	prec.	rec.	F1	prec.	rec.	F1
Basque	0.471	0.254	0.330	0.443	0.429	0.435
Bulgarian	0.745	0.312	0.440	0.638	0.631	0.634
English	0.565	0.245	0.342	0.430	0.425	0.428
German	0.763	0.310	0.441	0.703	0.699	0.701
Maltese	0.465	0.229	0.307	0.402	0.400	0.401
Navajo	0.686	0.112	0.193	0.449	0.430	0.435
Spanish	0.664	0.183	0.287	0.579	0.560	0.569
Swedish	0.818	0.378	0.517	0.783	0.737	0.759
Average	0.659	0.252	0.357	0.553	0.539	0.545

Table 4: Post-shared task results using the full data set for selected languages. These results use LM-B with a threshold value of 0.3.