Morphological Predictability of Unseen Words using Computational Analogy

Rashel Fam and Yves Lepage*

IPS, Waseda University
2-7 Hibikino, Wakamatsu-ku, Kitakyushu-shi, 808-0135 Fukuoka-ken, Japan
fam.rashel@fuji.waseda.jp, yves.lepage@waseda.jp

Abstract. We address the problem of predicting unseen words by relying on the organization of the vocabulary of a language as exhibited by paradigm tables. We present a pipeline to automatically produce paradigm tables from all the words contained in a text. We measure how many unseen words from an unseen test text can be predicted using the paradigm tables obtained from a training text. Experiments are carried out in several languages to compare the morphological richness of languages, and also the richness of the vocabulary of different authors.

Keywords: Unseen words, Word predictability, Paradigm tables.

1 Introduction

The current trend in natural language processing is to extract knowledge from a training corpus, apply this knowledge to perform some task on a test set, and measure the performance. As many techniques are generally first developed for English, they take the typographic word as their basic processing unit. For tasks such as speech recognition or machine translation, the words known by the system constitute the vocabulary of the system. Unseen words, or out-of-vocabulary (OOV) words, or new words, become a problem. Unseen words are in fact of the same kind as hapaxes. Let us recall that hapaxes in any standard English text are estimated to represent between 30 % to 50 % of the vocabulary of the text (44 % on Part A of the British National Corpus) while being of course unfrequent (less than 0.2 % of the total number of words in the same corpus).

In this paper, we address the problem of the predictability of unseen words: given an unseen word, find all other words that may explain it. We consider computational analogy as a possible way of explaining unseen words. For instance, the unseen word *inexhaustivity*[1] may be explained by: $\mathit{active} : \mathit{inactivity} :: \mathit{exhaustive} : x \quad \Rightarrow \quad x = \mathit{inexhaustivity}$. In the present work, on the contrary

* This work was supported by a JSPS Grant, Number 15K00317 (Kakenhi C), entitled Language productivity: efficient extraction of productive analogical clusters and their evaluation using statistical machine translation.

[1] No occurrence in Part A of the British National Corpus. Not present in the Oxford Dictionary of English. As of July 2016, Google returns only 1,650 hits for this word.

Table 1. Examples of analogies in different languages illustrating different phenomena. The formalization used in this paper captures infixing, but not repetition and reduplication

Phenomenon	Language	Example
Repetition	Indonesian	*pasar* : *pasar-pasar* :: *kota* : *kota-kota* 'market' : 'markets' :: 'town' : 'towns'
Reduplication	Latin	*cado* : *cecidi* :: *pago* : *pepigi* 'I fall' : 'I fell' :: 'I conclude' : 'I concluded'
Infixing	Arabic	*kalb* : *kulaib* :: *masjid* : *musaijid* 'a dog' : 'dogs' :: 'a mosque' : 'mosques'

to, for instance, [10] or [5], we will not take into consideration the meaning of words, but concentrate on the formal aspect of the problem. We adopt a standard experimental protocol: we first train a model on some training data and then test our model against some test set. The model trained is the set of paradigm tables that can be extracted from the set of words contained in a corpus, here the training set. This set of paradigm tables is supposed to reflect the organization of the lexicon of a language to a certain extent. In our experiments, we rely on it to predict new words, or rather, to measure how many of the unseen words from a test set are predictable using the organization of the lexicon contained in a training set as given by the paradigm tables.

For that, in Sect. 2 we present a pipeline, which relies on already reported research based on computational analogy, to automatically produce paradigm tables from all the words contained in a training text. We measure how many unseen words from an unseen test text can be predicted using these paradigm tables. Again, as we rely only on a computational definition of formal analogies, we do not take meaning into account in this work. In Sect. 3, experiments are carried out in different languages from four continents. At the same time, we compare one author against other authors reporting similar events, namely the Gospel of Luke against the three other ones. In this way, we attempt at characterizing the morphological richness of these languages as well as the richness of vocabulary of one author compared to other ones.

2 Pipeline for the Production of Paradigm Tables

The following pipeline relies on the notion of computational analogy between strings of symbols proposed in [3]. The definition that we use here exploits a specific notion of ratio between two strings of symbols to first build analogical clusters which are then merged into paradigm tables.

2.1 Extracting Analogical Clusters

We firstly define the ratio between two words A and B as a vector of features made of all the differences in number of occurrences in the two words, for all

the characters, whatever the writing system[2]; plus the distance between the two words[3]. See definition in (1). The notation $|S|_c$ stands for the number of occurrences of character c in string S and $d(A, B)$, is the edit distance between two strings A and B. This definition of ratios captures prefixing and suffixing and more generally infixing. Infixing is mandatory for a proper treatment of semitic languages [11]. However, this definition does not capture reduplication nor repetition. The latter one would be needed to capture marked plurals. Examples of different phenomena are listed in Table 1.

$$A : B \quad \triangleq \quad \begin{pmatrix} |A|_a - |B|_a \\ \vdots \\ |A|_z - |B|_z \\ \mathrm{d}(A, B) \end{pmatrix} \tag{1}$$

Based on the notion of ratio, we then define an analogy, more precisely a proportional analogy of commutation between strings of symbols, as a relationship between four objects where two properties are met: (a) equality of ratios between the first and the second terms on one hand and the third and the fourth terms on the other hand, and (b) exchange of the means. The exchange of the means states that the second and the third terms can be exchanged. The notation and the definition of an analogy are given in (2) at the same time[4].

$$A : B :: C : D \qquad \overset{\Delta}{\iff} \qquad \begin{cases} A : B = C : D \\ A : C = B : D \end{cases} \tag{2}$$

From the entire set of words contained in a text, we compute the set of analogical clusters, i.e., series of word pairs in which any two word pairs is a proportional analogy as defined in (2). Such analogical clusters are defined in (3). Notice that the order of word pairs in analogical clusters has no importance.

$$\begin{array}{l} A_1 : B_1 \\ A_2 : B_2 \\ \quad \vdots \\ A_n : B_n \end{array} \quad \overset{\Delta}{\iff} \quad \forall (i, j) \in \{1, \ldots, n\}^2, \quad A_i : B_i :: A_j : B_j \tag{3}$$

To produce the set of analogical clusters, we first group pairs of words by equal ratio in number of characters using the method proposed in [4]. The complexity is $O(n^2)$ in the worst case with n the number of words. We then test for

[2] Taken from the characterizations of proportional analogy of commutation in [3, 9].

[3] Taken from the characterization of proportional analogy of commutation in [3]. The only two edit operations involved are insertion and deletion. The purpose is to indirectly take into account the number of common characters appearing in the same order in A and B because $d(A, B) = |A| + |B| - 2 \times s(A, B)$ where $|S|$ denotes the length of string S and $s(A, B)$ the length of the longest common sub-sequence (LCS) between A and B.

[4] Trivially, $|A|_a - |B|_a = |C|_a - |D|_a \quad \Leftrightarrow \quad |A|_a - |C|_a = |B|_a - |D|_a$. Hence, the equalities on features added by $A : C = B : D$ in (2) in fact reduces to one: $d(A, C) = d(B, D)$.

equality between distance for each word pair. This may split the sets of word pairs in smaller sets of word pairs for which all word pairs have the same ratio. Finally, for each such set of word pairs with equal ratio, we test for equality in edit distance vertically, i.e., we verify that $A_i : A_j = B_i : B_j$ for any pair of word pairs (i, j) (see Footnote 4). Cases where the equality is not met lead to split the set into smaller sets. Ideally, this is equivalent to extract all maximal cliques in the undirected graph whose set of vertices is a word pair i and where there is an edge between word pair i and word pair j if and only if the constraint $A_i : A_j = B_i : B_j$ is met. Existing algorithms for this problem [1] are time-consuming. For this reason, we adopt a heuristic which does not ensure that all maximal cliques are output, but ensures that all nodes belong to one of the maximal cliques output (see Algorithm 1).

We ensure that any two word pairs in a series of word pairs of equal ratio, say, A, B and C, D, also verifies $A : C = B : D$.

2.2 Producing Paradigm Tables

Individual analogical clusters already give some insight at the organization of the lexicon. Paradigm tables [8, 6, 2] give a more compact view by merging several analogical clusters. A paradigm table is a matrix of words where four words from two rows and two columns are an analogy (4). As the order on rows and columns is indeed not relevant, one should think of a torus in three-dimensional space, rather than a matrix in two dimensions.

$$\begin{array}{c} P_1^1 : P_1^2 : \cdots : P_1^m \\ P_2^1 : P_2^2 : \cdots : P_2^m \\ \vdots \quad \vdots \qquad\quad \vdots \\ P_n^1 : P_n^2 : \cdots : P_n^m \end{array} \quad \overset{\Delta}{\Longleftrightarrow} \quad \begin{array}{l} \forall (i,k) \in \{1,\ldots,n\}^2, \\ \forall (j,l) \in \{1,\ldots,m\}^2, \quad P_i^j : P_i^l :: P_k^j : P_k^l \end{array} \tag{4}$$

We create paradigm tables from analogical clusters as follows. A paradigm table is first initialized from one analogical cluster and then expanded by adding other analogical clusters to it. There are two possible ways of adding a cluster to a paradigm table. In the first case, if a column in the paradigm table shares at least three words with a column in an analogical cluster, this cluster can be added vertically to it. In the second case, an analogical cluster shares more than three words on a row of the paradigm table, so that the cluster can be transposed and inserted to the paradigm table horizontally.

Algorithm 2 sketches the necessary functions for the production of paradigm tables from analogical clusters. In these functions, the strategy is to process longer analogical clusters first because the possibility of inserting smaller new series in a paradigm table increases with the number of words it contains. To ensure that no insertion is forgotten, the list of series of word pairs of equal ratio is scanned several times. However, because clusters are added only to one paradigm table, the complexity is $O(n^2)$ in the worst case with n the number of clusters.

Algorithm 1 Building a set of analogical clusters from a set of words

function BUILD_CLUSTERS(*set of words*)
  tree ← from the *set of words* ▷ Hierarchically group words by their ▷ number of occurrences of characters.
  repeat top-down exploration of the *tree* against itself
    group pairs of words by equal difference
      of number of occurrences of characters
  until last character
  for all *set of word pairs* with equal number of occurrences of characters **do**
    CHECK_DISTANCE(*set of word pairs*)
  end for
end function

function CHECK_DISTANCE(*set of word pairs* $(A_1, B_1), \ldots, (A_n, B_n)$)
  for all $i \in \{1, \ldots, n\}$ **do**
    compute $\mathrm{d}(A_i, B_i)$
  end for
  for all *set of word pairs* (A_i, B_i) with same distance **do**
    CHECK_CLUSTER(*set of word pairs*)
  end for
end function

function CHECK_CLUSTER(*set of word pairs* $(A_1, B_1), \ldots, (A_n, B_n)$)
  $\mathcal{V} \leftarrow \{1, \ldots, n\}$ ▷ Vertices of the graph.
  $\mathcal{E} \leftarrow \{(i, j) \in \mathcal{V}^2 \ / \ A_i : A_j = B_i : B_j\}$ ▷ Edges of the graph.
  list ← *nodes* in $\mathcal{V}$ sorted by non-increasing number of edges
  not_yet_covered ← $\mathcal{V}$
  repeat
    i ← first node in *list*
    delete i from *list*
    if $i \in$ *not_yet_covered* **then**
      clique ← $\{i\}$ ▷ Initialize clique to singleton of not yet explored vertex.
      clique, *not_yet_covered* ← EXPAND_CLIQUE(*clique*, *list*, *not_yet_covered*)
      return *clique* ▷ *clique* is an analogical cluster.
    end if
  until *not_yet_covered* $= \emptyset$
end function

function EXPAND_CLIQUE(*clique*, *list*, *not_yet_covered*)
  for all i in *list* **do**
    if i is connected with all vertices in the *clique* **then**
      add i to the *clique* ▷ Remains a clique.
      delete i from *not_yet_covered*
    end if
  end for
  return *clique*, *not_yet_covered*
end function

It is worth noticing that, when creating all possible paradigm tables from a text, not all of the words will necessarily appear in a paradigm table. Reciprocally, paradigm tables extracted from texts may contain blank cells. A paradigm table which does not contain any blank cell is not productive as no new word can be entered in it. On the contrary, we will call any paradigm table which contains at least one blank cell a *productive paradigm table*. We will call a word that may fill a blank cell in a productive paradigm table a *predictable word*.

We simply define the size of a paradigm table as its number of rows multiplied by its number of columns. We measure the density of a paradigm table as the ratio of non-blank cells over the total number of cells, i.e., the size of the paradigm table. With this definition a non-productive table has a density of 100 %. *Productive* paradigm tables have a density *less than* 100 %.

In the experiments reported hereafter, we monitor the density of the paradigm tables produced by controlling the addition of analogical clusters: we add a cluster to a paradigm table only if the density of the new paradigm table after adding the cluster is above a given threshold. This is done by the condition in the function EXPAND_TABLE in Algorithm 2.

Algorithm 2 Building a set of paradigm tables from a set of analogical clusters

```
function BUILD_PARADIGM_TABLES(set of analogical clusters, threshold)
    tables ← ∅                                  ▷ Set of paradigm tables, initially empty.
    list ← set of analogical clusters sorted by non-increasing order of size
    repeat
        analogical cluster ← first analogical cluster in list
        delete analogical cluster from list
        table ← analogical cluster              ▷ Make analogical cluster a paradigm table.
                                                ▷ By construction, it has only 2 columns
                                                ▷ and a density of 100 %.
        table, list ← EXPAND_TABLE(table, list, threshold)
        tables ← tables ∪ {table}
    until list is empty
    return tables
end function

function EXPAND_TABLE(table, list, threshold)
    repeat                                      ▷ Possibly scan the list several times.
        for all cluster in the list (in non-increasing order of sizes) do
            if cluster can be added to table and density of new table ≥ threshold then
                add cluster to table (either transposed or not)
                delete cluster from list
            end if
        end for
    until no cluster can be added to table
    return table, list
end function
```

3 Experiments

3.1 Languages and Texts Used

We selected several languages from four continents. We chose three languages per continent. The choice of a language over another was first driven by the availability of the texts themselves, the availability of a locale for pre-processing and the confidence that we had that segmentation into words, when needed, did not go wrong. We also tried to represent different linguistic families as much as possible. The selected languages are the following ones.

Africa: Somali (so), Swahili (sw), Xhosa (xh);
America: Achuar (acu), Nahuatl (nah), Quichua (qu);
Asia: Chinese (zh), Indonesian (id), Telugu (te);
Europe: English (en), Finnish (fi), (Modern) Greek (el).

The texts we use in our experiments are texts available in a relatively large number of languages: they are translations of the New Testament collected by Christodoulopoulos[5]. We insist on using the same text in all languages so as to ensure reliable observations and comparisons across languages to a certain extent. We use Matthew's Gospel as training data to produce paradigm tables. We use Luke as our test set, i.e., we shall extract all words from Luke which do not appear in Matthew and examine whether these words are predictable from the organization of the vocabulary obtained from Matthew.

Table 2 gives statistics on the number of words in each language. The training and the test sets are similar in subject and size. The numbers of tokens (all words) and types (different words) are slightly higher in the test set than in the training set. It is the same for the type–token ratio (with the exception of Quichua) but the same variations across languages are observed

3.2 Statistics on the Productive Paradigm Tables Produced

Table 3 shows the number of productive paradigm tables produced in each language for two different thresholds, 50 %, and 90 %. Let us first recall that we controlled the density of the paradigm tables during their production. Second, as we are interested in predicting new words, we left out non-productive paradigm tables. Productive paradigm tables have a density such that: threshold $\leq$ density $<$ 100. Their number varies considerably across languages, from 220 (Chinese) to 11,349 (Achuar) for a threshold of 50 % (115 (English) to 2,783 (Achuar) for a threshold of 90 %). Their average size is relatively stable around 55 for 50 %, with Chinese an outlier at 88 (same observation for the threshold of 90 % with a size of 14, Chinese behaving like other languages this time). For the threshold of 50 %, an average size of 55 may be interpreted as 7 rows by 8 columns, with half of empty cells empty (3 rows by 4 columns with one blank cell for the threshold of 90 %). However, finer observation shows that the distribution of the sizes of paradigm tables in one language is not Gaussian.

[5] `http://homepages.inf.ed.ac.uk/s0787820/bible/`. This corpus is a continuation of previous efforts described in [7].

Table 2. Statistics on the training and test sets used in each language

Language	Training set (Matthew)			Test set (Luke)		
	Number of tokens	Number of types	Type-token ratio (%)	Number of tokens	Number of types	Type-token ratio (%)
Achuar	22,470	5,349	23.8	23,177	5,609	24.2
Chinese	18,350	4,030	22.0	19,956	4,488	22.5
English	23,726	2,098	8.8	25,987	2,370	9.1
Finnish	17,331	4,467	25.8	18,804	5,003	26.6
Greek	20,438	3,819	18.7	21,856	4,367	20.0
Indonesian	22,375	2,450	10.9	23,623	2,650	11.2
Nahuatl	23,222	3,833	16.5	24,060	4,096	17.0
Quichua	15,038	4,066	27.0	16.332	4,249	26.0
Somali	20,375	3,967	19.5	21,535	4,244	19.7
Swahili	16,851	3,926	23.3	18,467	4,411	23.9
Telugu	13,083	6,066	46.4	14,404	6,747	46.8
Xhosa	14,505	5,580	38.5	15,537	6,265	40.3

The product of the first three columns in Table 3 directly gives the number of non-blank cells in the paradigm tables, and indirectly, the number of blank cells. For instance for English, with a threshold of 50 %, we have $587 \times 49.5 \times (100-58.3)/100 = 12,117$ blank cells. This is the number of words which are predictable from the organization of the vocabulary given by the paradigm tables. It is of course natural that the number of unseen words which can actually find a place in the paradigm tables be lower for a higher density threshold, in absolute numbers. For instance, for English we observe a decrease from 12,117 predictable words for a threshold of 50 % to only $115 \times 12.6 \times (100 - 91.8)/100 = 119$ predictable words for a threshold of 90 %.

3.3 Predicting Unseen Words using Productive Paradigm Tables

We now turn to the experiments in filling paradigm tables with unseen words. The results are also given in Table 3. The number of unseen words in the different languages ranges from a little bit less than 1,000 (English and Indonesian) to almost 5,000 words (Telugu). This reflects a known fact: Luke would have a richer vocabulary than the other Evangelists, and this seems to have been carried over in translation. In Table 3, these numbers are repeated for the two thresholds.

As for prediction, it is natural to expect that the number of predicted unseen words would decrease for a higher threshold of density, because the number of blank cells in the paradigm tables is smaller. Table 3 shows this phenomenon (Ratio of Unseen words): a reduction from 15 % to around 1 % of predicted unseen words is observed in average. This is the ratio of words from Luke, which were unseen in Matthew but can fill in a blank cell in some of the paradigm tables.

Table 3. Statistics on productive paradigm tables produced from the training set with density thresholds of 50% (top) and 90 % (bottom) and predicted unseen words from the test set in each language

	Productive paradigm tables			Unseen words		
Language	Total number	Avg size	Avg density (%)	Total number	Predicted	Ratio (%)
Achuar	11,349	49.1	53.4	2,801	748	26.7
Chinese	220	88.4	55.9	2,497	193	7.7
English	587	49.5	58.3	858	75	8.7
Finnish	2,147	49.7	57.6	2,597	331	12.7
Greek	793	64.1	57.7	2,238	352	15.7
Indonesian	790	48.3	57.8	940	126	13.4
Nahuatl	512	67.9	57.1	2,143	296	13.8
Quichua	4,478	59.3	55.0	2,170	900	41.5
Somali	2,078	61.8	55.1	1,929	392	20.3
Swahili	2,067	53.6	56.5	2,381	430	18.1
Telugu	557	74.4	56.0	4,485	459	10.2
Xhosa	3,501	60.2	55.2	3,807	734	19.3
Achuar	2,783	13.8	91.7	2,801	90	3.2
Chinese	198	13.3	91.7	2,497	7	0.3
English	115	12.6	91.8	858	1	0.1
Finnish	611	14.7	91.7	2,597	31	1.2
Greek	530	17.7	91.5	2,238	61	2.7
Indonesian	178	12.7	91.7	940	4	0.4
Nahuatl	303	16.6	91.7	2,143	33	1.5
Quichua	2,728	17.8	91.5	2,170	217	10.0
Somali	1,411	17.7	91.6	1,929	76	3.9
Swahili	648	14.2	91.7	2,381	37	1.6
Telugu	316	16.4	91.7	4,485	34	0.8
Xhosa	2,209	15.6	91.6	3,807	142	3.7

Across languages, one observes variations which are not necessarily the same for the two thresholds. As one can interpret the ratio of 90 % to correspond to a safer production of new words, in a same language, the ratio (not given here) between the predicted unseen words for the two thresholds, gives the proportion of reliable words produced. This proportion is quite high for Quichua: $215/900 = 24\,\%$, while it is quite low for English: $1/75 = 1.3\,\%$. These figures characterize in some way the morphological richness of the languages.

4 Conclusion

In this paper, we presented a pipeline for the production of paradigm tables from words contained in a given text by relying on a formalization of analogy.

The blank cells in the produced paradigm tables stand for potential word forms. We carried out experiments to see how many of the words used by an author can be predicted from such paradigm tables in comparison to another author. The results obtained in a variety of languages of the world, with two different thresholds for the density of the paradigm tables produced, can be used to characterize relative morphological richness of languages as well as the richness of the vocabulary of authors.

References

1. Bron, C., Kerbosch, J.: Algorithm 457: finding all cliques of an undirected graph. Communications of the ACM 16(9), 575–577 (1973)
2. Hathout, N.: Acquisition of the morphological structure of the lexicon based on lexical similarity and formal analogy. In: Proceedings of the 3rd Textgraphs workshop on Graph-based Algorithms for Natural Language Processing. pp. 1–8. Coling 2008 Organizing Committee, Manchester, UK (August 2008), http://www.aclweb.org/anthology/W08-2001
3. Lepage, Y.: Lower and higher estimates of the number of "true analogies" between sentences contained in a large multilingual corpus. In: Proceedings of COLING-2004. vol. 1, pp. 736–742. Genève (Aug 2004)
4. Lepage, Y.: Analogies between binary images: Application to Chinese characters. In: Prade, H., Richard, G. (eds.) Computational Approaches to Analogical Reasoning: Current Trends, pp. 25–57. Springer, Berlin, Heidelberg (2014), http://dx.doi.org/10.1007/978-3-642-54516-0_2
5. Mikolov, T., Yih, W.T., Zweig, G.: Linguistic regularities in continuous space word representations. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2013). pp. 746–751. Association for Computational Linguistics, Atlanta, Georgia (June 2013), http://www.aclweb.org/anthology/N13-1090
6. Neuvel, S., Fulop, S.A.: Unsupervised learning of morphology without morphemes. In: Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning. pp. 31–40. Association for Computational Linguistics (July 2002), http://www.aclweb.org/anthology/W02-0604
7. Resnik, P., Olsen, M.B., Diab, M.: The Bible as a parallel corpus: Annotating the 'book of 2000 tongues'. Computers and the Humanities 33(1), 129–153 (1999), http://dx.doi.org/10.1023/A:1001798929185
8. Singh, R., Ford, A.: In praise of Sakatayana: some remarks on whole word morphology. In: Singh, R. (ed.) The Yearbook of South Asian Languages and Linguistics-200. Sage, Thousand Oaks (2000)
9. Stroppa, N., Yvon, F.: An analogical learner for morphological analysis. In: Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005). pp. 120–127. Association for Computational Linguistics, Ann Arbor, Michigan (June 2005), http://www.aclweb.org/anthology/W/W05/W05-0616
10. Tekauer, P.: Meaning Predictability in Word Formation: Novel, Context-free Naming Units. John Benjamins Publishing (2005)
11. Wintner, S.: Natural Language Processing of Semitic Languages, chap. Morphological Processing of Semitic Languages, pp. 43–66. Springer, Berlin, Heidelberg (2014), http://dx.doi.org/10.1007/978-3-642-45358-8_2