PARSING THE JSON FILES

The files I worked with to obtain the data for the data frames comes from the JSON files that are included with every ORACC text. They have a certain hierarchy which broadly divides each text into meaningful components (I assume) labelled “discourse” and “sentence,” but the meat of the information comes under the level of the “cdl,” which is under the node of type “sentence.” At this level is a list were the attribute “node” guides us, telling us what sort of information comes beneath it in the hierarchy. If the node value is “d,” it is a line label such as “o i 2.” If the node value is “l,” it designates a word. There used to be a node labelled “c,” which designated a phrase, but this was no longer there when I received the updated JSON files.

The words are divided into signs, and this is the level where I need to extract the data. There are three types of signs: syllable, logogram and number with the key “v”, “s”, and “n” respectively. For signs with a paleography variation, there is an addition “mods” key, which contains information about the sign’s form. Under this node are the keys “f”, “a”, and “m”. I think ideally, the key “f” is supposed to map to the type of difference, e.g. “m” for a missing wedge. It seems that “a” should only map to “v” for the by-form of the LU₂ sign and “m” should map to “v” for the by-form of the TA sign. I’m not sure why these need separate columns.

One must be careful when writing a parser for these JSON files for two main reasons. One reason is that their structure changes. As stated earlier, the node label “c” was omitted with the updated files. There were other minor changes within the structure as well, which forced me to alter my code. I may have to alter this code in the future if the structure changes again.

I avoided writing recursive code for this reason. I wanted to see in the code how the structure of the JSON was flowing. The only extra method I wrote was to extract the information for an individual sign node. The main parser code has some repetitions.

SELECTING FEATURES

In general, I figured that the distribution of one sign, syllable, or word among its variant forms would denote a feature. For instance, if the syllable /šu/ is written <šu> 15 times and written <šu₂> 5 times, the feature would be the tuple (.75,.25).

Why not Tf-Idf? This method would highlight the importance of a value as it is important compared to the other values in the same text and to its distribution among the other texts of the corpus. I’m not concerned with how a value compares to every other value in the text, only to its variant counterpart. For example, <šu> as a syllable is not important compared to every writing of every syllable, only to its variant <šu₂>. I’m also not concerned with how the usage of a value in one text compares to its usage in another text. The amount of usage of a syllable or sign in a text does not relate to the form that is being used to represent it. It the syllable itself shows up less often, then naturally one of its forms will also show up less often. I’m not measuring the usage of a syllable of a sign, rather how it is being represented.

The text matrix is then set up by counting up the instances of the variations of these features per letter. This raw count is useful in itself to examine as I’ll do later in the results section. However, the value of the features is not given by the raw count, which could skew the data in favor of longer letters, but by calculating the distribution as discusses above. This can cause an issue if there are no instances of any feature variant in a given letter. As a place holder, I assign the general distribution of these variants across the corpus where this occurs. Previously I tried smoothing the counts by adding 1 across the board, but I have abandoned this position because the raw counts per each letter are so low that adding 1 can significantly alter the result.

Now the trick is selecting which syllable, sign and word variations might make the most impact for clustering. For this I have three basic guidelines: 1) Look for variations in each of the three categories that have the highest usage throughout the corpus, 2) discover the contextual usage for these variations, i.e if one variant appears only in one context, we can argue that its usage is not based convention across the corpus rather than scribal preference or tendency, and 3) examining the text count matrix with different variations sets grouped by different classifications to see how these classes are grouping and based on what features.

1. This step involves counting up the most frequent potential features and variants. Here is a chart for sign form variants:

|  |  |  |
| --- | --- | --- |
| **sign\_form** | **combined** | **count** |
| **BA** | **BA:..** | 401 |
| **BA:.t.** | 120 |
| **BU** | **BU:..** | 960 |
| **BU:.p.** | 109 |
| **DA** | **DA:..** | 434 |
| **DA:.d.** | 201 |
| **DI** | **DI:..** | 1024 |
| **DI:.d.** | 156 |
| **LI** | **LI:..** | 465 |
| **LI:.d.** | 147 |
| **LU₂** | **LU₂:..** | 218 |
| **LU₂:v..** | 1147 |
| **LU₂:v.y.** | 500 |
| **NA** | **NA:..** | 2500 |
| **NA:.t.** | 442 |
| **NI** | **NI:..** | 4666 |
| **NI:.d.** | 281 |
| **RU** | **RU:..** | 444 |
| **RU:.d.** | 108 |
| **SU** | **SU:..** | 245 |
| **SU:.t.** | 116 |
| **TA** | **TA:..** | 960 |
| **TA:.m.v** | 309 |
| **TI** | **TI:..** | 278 |
| **TI:.t.** | 450 |
| **U₂** | **U₂:..** | 1302 |
| **U₂:.m.** | 325 |
| **|ME.U.U.U|** | **|ME.U.U.U|:..** | 1375 |
| **|ME.U.U.U|:.m.** | 565 |
| **ŠA** | **ŠA:..** | 1082 |
| **ŠA:.d.** | 263 |
| **ŠA:.dm.** | 291 |
| **ŠA:.m.** | 157 |
| **ŠA:.y.** | 110 |
| **ŠA:.yd.** | 124 |

Here’s a chart for syllable variants:

|  |  |  |
| --- | --- | --- |
| **str\_part** | **b** | **count** |
| **aš** | **aš** | 392 |
| **aš₂** | 427 |
| **ia** | **ia** | 1724 |
| **ia₂** | 424 |
| **li** | **li** | 701 |
| **li₂** | 1191 |
| **tu** | **tu** | 440 |
| **tu₂** | 533 |
| **u** | **u** | 1503 |
| **u₂** | 1843 |
| **u₃** | 256 |
| **ša** | **ša** | 2827 |
| **ša₂** | 439 |
| **šu** | **šu** | 747 |
| **šu₂** | 1725 |

And a chart for word variants:

|  |  |  |
| --- | --- | --- |
| **lemma\_norm** | **lemma\_norm\_form** | **count** |
| **bēlu[lord]N:bēlī** | **bēlu[lord]N:bēlī:EN** | 182 |
| **bēlu[lord]N:bēlī:be-li₂** | 642 |
| **bēlu[lord]N:bēlīya** | **bēlu[lord]N:bēlīya:EN-a** | 76 |
| **bēlu[lord]N:bēlīya:EN-ia** | 770 |
| **bēlu[lord]N:bēlīya:EN-ia₂** | 190 |
| **bēlu[lord]N:bēlīya:be-li₂-ia** | 309 |
| **bēlu[lord]N:bēlīya:be-li₂-ia₂** | 194 |
| **ištu[from]PRP:issu** | **ištu[from]PRP:issu:TA** | 113 |
| **ištu[from]PRP:issu:TA@v** | 346 |
| **libbu[interior]N:libbi** | **libbu[interior]N:libbi:ŠA₃** | 217 |
| **libbu[interior]N:libbi:ŠA₃-bi** | 277 |
| **lā[not]MOD:lā** | **lā[not]MOD:lā:la** | 623 |
| **lā[not]MOD:lā:la-a** | 117 |
| **lū[may]MOD:lū** | **lū[may]MOD:lū:lu** | 516 |
| **lū[may]MOD:lū:lu-u** | 190 |
| **muhhu[skull]N:muhhi** | **muhhu[skull]N:muhhi:UGU** | 681 |
| **muhhu[skull]N:muhhi:UGU-hi** | 56 |
| **mā[saying]PRP:mā** | **mā[saying]PRP:mā:ma** | 61 |
| **mā[saying]PRP:mā:ma-a** | 1442 |
| **māru[son]N:mār** | **māru[son]N:mār:DUMU** | 61 |
| **māru[son]N:mār:{LU₂}A-KIN** | 51 |
| **pānu[front]N:pān** | **pānu[front]N:pān:IGI** | 142 |
| **pānu[front]N:pān:pa-an** | 146 |
| **šulmu[completeness]N:šulmu** | **šulmu[completeness]N:šulmu:DI-mu** | 597 |
| **šulmu[completeness]N:šulmu:šul-mu** | 91 |
| **šū[he]IP:šû** | **šū[he]IP:šû:šu-u** | 52 |
| **šū[he]IP:šû:šu-u₂** | 81 |
| **ṣābu[people]N:ṣābāni** | **ṣābu[people]N:ṣābāni:ERIM-MEŠ** | 67 |
| **ṣābu[people]N:ṣābāni:{LU₂}ERIM-MEŠ** | 148 |

2. Mixed vs. Complementary Distribution. If the usage of these variants is context-dependent, meaning that one form or syllable is used in one context and another form or syllable in another context, it does not tell us much about the preferential usage of the signs. This is known as a complementary distribution. For example, if a scribe uses *li2* only in the form of the word be-li2 and the *li* sign in all other contexts, the choice of sign usage is not determined by the scribe's preference rather on scribal convention. This convention would thus be utilized by every scribe of this corpus and not help us to detect subgroups among these texts where scribes differ.

On the other hand, if sign form or syllable variants appear within the same contexts, it gives us the information we want on scribal writing preference or tendencies. For example, *ia* and *ia2* both appear in forms of the word bēliya, meaning that a scribe had an option of orthography and incised one or the other. The question then becomes whether certain letters group together based on their tendencies to use one variant within a mixed distribution versus another variant.

3. This selection method involves looking at the feature counts for each letter grouped by different evaluation standards (to be discussed later).

CLUSTERING

Overview

The method of clustering we are using is K-Means. K-Means is a hard clustering algorithm which takes a set of vectors and groups them into a fix number of clusters. The basic idea is to initially place random cluster centers and measure the distance between the text vectors and the centers. A text vector is considered part of a cluster if it is closest to its cluster center. The centers are then reset as the average of all text vectors in its cluster and the distances and closeness is measure again. This process continues until the next iteration does not alter the clustering or a predetermined number of iterations is set. The RSS or residual sum of squares is the total distance of each text vector to its cluster center. For a good clustering, this value should be minimal.

A challenge for this algorithm is determining the number of clusters, K. One method for doing so is by using the elbow method. This method takes a measure of the RSS for incremental values of K. If you chart these two variables, you should see a significant bend (elbow) in the graph where an additional cluster does not add much of a benefit i.e. the RSS does not decrease by very much. This lets you know that the number of clusters is a useful number.

Another method for measuring the significance of the clustering is the silhouette score. This score measures how well the text vectors fit within their own cluster compared to their neighboring cluster.

Coding

Given the options for the features, I wanted to see which features or set of features would provide the best clustering. To do this I set up lists of feature pairs and took their powerset. I then iterated over each member of the powerset with different numbers of clusters so I could evaluate the quality of the clustering using the elbow method and the silhouette score.

From an initial look at the results, it seems that clustering works very well if one feature pair is selected with a K between 3-6. Once other feature pairs are included, the elbow in the chart becomes lost. This would seem to indicate that the features are not clustering in the same way, i.e. there is no interdependence between the features. This is unfortunate if we want to group certain features together. This issue comes up late in the plene writing section.

I wanted to make a note here about using K-Means instead of other clustering algorithms. David gave me the suggestion of using naïve-Bayes because certain features were not existent in some texts and K-means requires that values be given to them. I have thus far gotten around this by assigning the overall distribution of features to those letters where the feature is absent. As I’ve read up on naïve-Bayes I am not sure how this method ignores the absent features. Naïve-Bayes calculates a class for the letter by selecting one that has the highest probability of containing a vector over all other classes. Part of the calculation involves computing the product of the probabilities that any feature in one letter fits into that feature across a class. If the feature is absent from a letter, then what value is given for that component of the equation? If we claim to simply ignore it, that means we are effectively giving that probability a value of 1. Another option I can think of is to apply a substitute value of the general distribution across the corpus and calculate the probability of that value among the class distribution of the feature. This is essentially the same thing that I’ve done for K-means. I’m still uncertain of the benefits of using naïve-Bayes.

This section on clustering does not require any information about the letters themselves such as who sent them and whence they were sent. This is needed for the next section on evaluation.

EVALUATION

Before you can evaluate your clusters you need to select a classification for the texts. This allows you to compare the clusters that the algorithm selects to a standard that is selected by a parameter expected by an expert in the subject. For this corpus we are thus far examining three types of classification: 1) sender location, 2) dossier, and 3) SAA Chapter. The evaluation will measure how well our clusters correspond to each class. For example, how well do letters from Arrapha fit into one cluster and letters from Manzamua fit into another.

We employ an evaluation metric called the purity score to measure this cluster/class fit. To calculate the purity score, each cluster is examined and the number of items from the class that appears most in the cluster is tallied. The sum of all these fits is taken and divided by the total number of letters in the corpus, thus making the ideal purity score 1.

Even if a high number for the purity score is not achieved there may still be some usefulness to the clustering. We should examine how well the purity score for the clustering compares to the purity score for a random clustering. If the purity scores are similar, we know our classification is a poor way to group the data. If the purity score for our clustering is significantly higher than the purity score for the random clustering, more can be said.

There are 60 certain sender locations, 161 unique dossiers (including all uncertainty), and 37 total SAA Chapters. The certainty of the dossiers is an issue because if we limit the corpus to only certain dossiers (those marked with “.a” at the end), there are only 276 letters with 106 unique dossiers. This would give us an artificially high purity score by the ratio of number of groups to texts in corpus alone. Including “.b” endings does not improve matters adding only 69 letters. The majority of letters in the corpus (609) have “.c” level uncertainty. My concern is that we don’t have enough certain dossiers to say anything meaningful that we might compare to the uncertain dossier letters.

Coding

I programmed in a similar way to the clustering. I iterated through the powerset of features and measured purity scores for each of the three types of classification. In terms of raw purity score, the dossier classification performed the best with the highest score at around .28. This is not surprising given that high numbers of classes naturally improve purity scores. Then sender location peaked at .23 purity followed by SAA chapter with a high of around .21. Within each class type, the word variation features had highest purity score followed by paleography and then syllable features.

In terms of the difference between random purity and cluster purity the results were slightly different. Here, the SAA chapter yielded the highest difference followed by sender location and then dossier, the highest difference being around .08. The same order was maintained with type as the raw purity score. The significance of this still eludes me as present

FURTHER EXAMINATIONS

PLENE WRITING

Another example of a feature to distinguish in a letter is the usage of plene vs. non-plene writings. A plene writing is an extra sign written to highlight the vowel of a previous CV-sign or the next VC-sign. The two signs would thus be CV-V or V-VC where the vowels are equivalent. Such a writing is considered “extra” or plene because it is not strictly necessary in a syllabic writing system, but is nevertheless used, usually when a vowel is long. The distribution of the usage of plene writings vis a vis their oppositional counterparts (CV-V or CV) could, like other features, indicate a particular scribe’s preference.

One reason to consider plene writings as a feature for clustering is a comment made by Parpola for LAS 14 (1983:19). There, he proposes the possibility that this letter was written by Akkullānu given his preference in other letters to write in a plene style. If clustering based on this feature yields significant results, it may validate this argument for authorship of a letter.

However, one must be careful when applying this argument across any oppositional pair. Certain words have a preference across the corpus to be written one way instead of the other. For example, the word *mā*, a particle introducing direct speech, has a major tendency (96% of the time) to be written in the plene form <ma-a> as opposed to <ma>. In contrast the words *lā*, a negative particle, and *lū*, a precative particle, have a bend toward the non-plene writing (84% and 73%). Therefore, the plene writings of <ma-a>, <la-a>, and <lu-u> should not be combined into one feature because if even if a scribe prefers to write without extra V-signs, they may still write <ma-a> due its heavy expectation. It may be possible to get around this problem by excluding words like *mā*. (This remains to be done)

Coding

The method of finding plene writings in corpus requires some work. I composed a regular expression to select the word forms that contain a V-sign copy of the previous or following sign’s vowel and filtered the data frame on the *form* column. This only gave me the possible words (lemma, normalization, and form) that could have a plene writing. The next step was to find their non-plene counterparts. (more needed here)

RESULTS

TEST SCENARIOS

The following is an analysis of certain test scenarios described in the overview of dossiers. No clustering or evaluation is needed for this assessment, but the feature selection is examined.

**Nabu-belu-kaʾʾin in dossiers 15.02.01 and 15.04.01**

15.02.01 has 23 total letters

Higher usage of NA/t vs. NA (20:59 = 25.3% cf. 14.7% overall)

Regular usage of MEŠ/m vs. MEŠ (16:34 = 32% cf. 27.5% overall)

Higher usage of ia₂ vs. ia (31:21 = 40.4% cf. 19.7% overall)

Higher usage of ša₂ vs. ša (22:68 = 24.4% cf. 13.4% overall)

Higher usage of lu-u vs. lu (10:9 = 52.6% cf. 26.9% overall)

15.04.01 has 7 total letters

0 usage of NA/t (26 of NA)

0 usage of MEŠ/m (15 of MEŠ)

Higher usage of ia₂ vs. ia (7:6 = 53.8% cf. 19.7% overall)

1 usage of ša₂ (44 of ša)

Too few instances of lū to say much

There are 3 features which mismatch between these two dossiers (NA, MEŠ, ša). Either these features are good features and we cannot claim that these two dossiers were composed by the same scribe and thus the scribe did not travel with Nabu-belu-kaʾʾin but he gained a new scribe in his new location, or these features are not good features to cluster letters

**Shamash-belu-usur in dossiers 05.11.01 and 15.05.01**

05.11.01 has 10 total letters

0 usage of MEŠ/m

otherwise every feature seems to correspond to typical usage

15.05.01 has 18 total letters

Regular usage of MEŠ/m vs. MEŠ (14:30 = 31.8% cf. 27.5% overall)

Higher usage of šu vs. šu₂ (39:30 = 56.5% cf. 30.2%)

Higher usage of be-li₂-ia vs. EN-ia (16:4 = 80% cf. 28.6% overall)

Higher usage of lu-u vs. lu (8:7 = 53.3% cf. 26.9% overall)

There are 4 mismatching features (MEŠ, šu, bēliya, lū). See previous for conclusions.

**Calah under two governors, Marduk-remanni and Ashur-bani, in dossiers 01.05.01 and 01.05.02**

01.05.01 has 1 letter

01.05.02 has 13 letters

Skip for now

**Der under Il-yadaʾ (15.06.01) and Šamaš-belu-uṣur (15.05.01)**

15.06.01 has 22 letters

Regular/Lower usage of šu vs. šu₂ (16:83 = 16.2% cf. 30.2% overall)

Higher usage of ia₂ vs ia (38:67 = 36.2% cf. 19.7% overall)

Higher usage of be-li₂-ia vs. EN-ia (28:4 = 87.5% cf. 28.6% overall)

Higher usage of lu-u vs. lu (8:10 = 44.4% cf. 26.9% overall)

15.05.01 has 18 letters

Breakdown given already

Regular usage of ia₂ vs. ia (12:37 = 24.5% cf. 19.7% overall)

They match in irregular distributions of 2 features (bēliya, lū) but mismatch in 2 other features (šu and ia)

**Kar-Sharrukin**

15.04.02 has 16 letters

Higher usage of EN vs. be-li₂ (8:12 = 40% vs. 22%)

Regular usage for everything else

Compare to the 15.04.01 dossier for Nabu-belu-kaʾʾin, which also had mostly regular usage of all features except for the syllable /ia/. However there are few instances of that syllable in 15.04.01 and also few instances of bēlī in 15.04.02 to say anything more definitive. They seem to correspond pretty well by this metric.

**Mazamua, different offices**

15.05.02 has 7 letters

Higher usage of ia₂ vs. ia (16:25 = 39% cf. 19.7% overall)

Higher usage of šu vs. šu₂ (17:15 = 53.1% cf. 30.2% overall)

Higher usage of be-li₂-ia vs. EN-ia (11:0 = 100% cf. 28.6% overall)

2 matches among irregulars (šu and bēliya) but 2 mismatches (ia and lū)

15.05.03 has 4 letters

15.05.04 has 2 letters