International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 4 (2012) pp. 219 -227

Clustering Techniques for Establishing Inflectionally Similar Groups of Stems

Zacharias Detorakis [1], George Tambouratzis [2]

[1] Inst. for Language and Speech Processing, 6 Artemidos & Epidavrou Str.
Paradissos Amaroussiou, 151 25, Greece
zdetor@ilsp.gr

[2] Inst. for Language and Speech Processing, 6 Artemidos & Epidavrou Str.
Paradissos Amaroussiou, 151 25, Greece
giorg_t@ilsp.gr

***Abstract*: This article presents a hierarchical clustering algorithm aimed at creating groups of stems with similar characteristics. The resulting groups (clusters) are expected to comprise stems belonging to the same inflectional paradigm (e.g. verbs in passive voice) in order to support the creation of a morphological lexicon. A new metric for calculating the distance between the data objects is proposed, that better suits the specific application by addressing problems that may occur due to the limited amount of information from the data. A series of experimental results are provided, that demonstrate the performance of the algorithm, compare different distance metrics in terms of their effectiveness and assist in choosing appropriate approaches for a number of parameters.**

***Keywords*:** Agglomerative clustering, Hamming distance, inflectional paradigm, cluster proximity, cluster validity**.**

I. Introduction

A morphological lexicon is a lexicon that contains all different word-forms (words) that are generated for each lemma of a language. The main goal of the research presented in this article is to automatically create such a lexicon based on the terms that are found in a corpus. These terms refer to the results of the analysis of the word forms within a given text into stems and endings. The manual creation of a morphological lexicon is a tedious task that becomes more difficult for languages such as Modern Greek because of their highly inflectional morphology. Morphological lexica are of particular importance since they can be exploited in several natural language processing applications [1] such as search engines, information retrieval, machine translation systems, etc.

As an initial step, a stemmer has already been created for Greek, based on the concept of genetic algorithms ([2], [3]). This stemmer manages to recognize the stem of a word and distinguish it from its inflectional suffix with a high level of accuracy (namely 96% for 213,000 words). This level of accuracy has been calculated by comparing the experimental results of a GA-based approach to the contents of a handcrafted morphological lexicon created by a team of specialized linguists over a period of approximately 5 years at ILSP [4].

Most of the stemmers described in literature are used for information retrieval; therefore they achieve their purpose once they succeed in reducing a word to its stem. On the other hand, when trying to automatically create a morphological lexicon, one should be able to discover the underlying connection between different stems (i.e. stems belonging to the same grammatical category, gender, tense, etc.) and form groups, each of which contains a collection of stems with related characteristics and the same set of associated suffixes. These groups are referred to in literature as inflectional paradigms.

Linguistica, developed by Goldsmith [5], is one of the morphological analyzers proposed that in an effort to identify stems and other inflectional morphemes, groups together certain stems that share the same signature i.e. the same set of suffixes. Even though Linguistica doesn't ultimately obtain a single partition of the input data (i.e. the set of stems), the approach is similar to the creation of clusters. Table 1 illustrates the aim of clustering stems for the Greek language:

Table 1. Example of words and their inflectional suffixes

Lemma	English translation	Inflectional Suffixes
βάφω	paint (verb)	-ω, -εις, -ει, -ουμε, -ετε, -ουν
παίζω	play (verb)	-ω, -εις, -ει, -ουμε, -ετε, -ουν
βοηθώ	help (verb)	-ώ, -άς, -ά, -άμε, -άτε, -άν
μιλώ	speak (verb)	-ώ, -άς, -ά, -άμε, -άτε, -άν

All four stems belong to the general grammatical category of verbs; however the first two are attributed to a different inflectional paradigm (which represents one cluster) than the third and the fourth stem (which represent a second cluster) as indicated by the corresponding suffixes. Additionally, the example illustrates that such a clustering of stems is only necessary in highly inflectional languages, i.e. the corresponding English stems are all combined with the same set of suffixes and therefore should be grouped together.

To address the task, the system developed must perform a clustering of the stems (data objects) identified. A general flow chart indicating the processes that are applied for the creation of a morphological lexicon from a given corpus are illustrated in Figure 1.

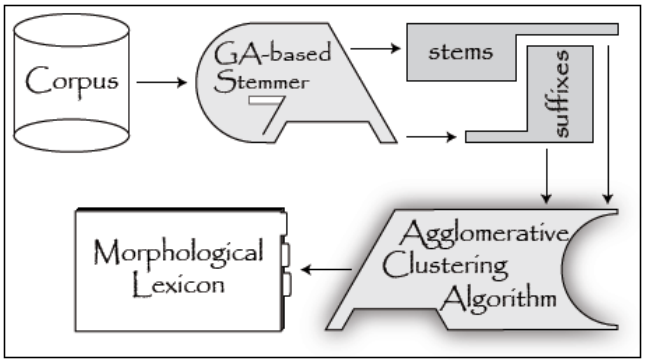

Figure 1. The main processing steps required to automatically generate a morphological lexicon from a corpus

There exist two main groups of clustering algorithms: partitional methods and hierarchical methods [6]. Partitional methods divide a set of data into smaller subsets, called clusters, by assigning each data object to exactly one subset. The simplest and most commonly used representative of this kind of clustering is the K-means algorithm [7]. K-means starts by choosing K initial centroids, where K is the desired number of clusters. Each object is thereafter assigned to its closest centroid and thereby K clusters are created. The centroids of the clusters are updated and the process of assigning objects to centroids is repeated iteratively until the centroids settle and all objects remain assigned to the same clusters in subsequent iterations. The main limitation that prohibits the use of K-means in the current application is that it requires the number of clusters to be specified a priori, which in this case cannot be predicted reliably. Moreover, even if the number of clusters was known, the K-means algorithm would not be suitable for this kind of data object because, as will be detailed in the following sections, there is no straightforward notion of a centroid in the given pattern space. Even though the K-medoid algorithm [8], which is related to K-means, is designed to overcome this restriction, it is not clear how it can process the data objects under study. Finally, any partitional clustering algorithm would fail to recognize and sufficiently represent the structure of the data, such as clusters containing subclusters which in turn can be further divided.

Therefore the search for methods for clustering together stems has focused on hierarchical clustering methods. This group of clustering methods is able to represent the taxonomy of the data in a tree-like structure called dendrogram. Hierarchical clustering methods don't require any priori knowledge of the number of clusters, other than setting a threshold value in order for the algorithm to reach a final clustering result.

II. Stemming Based on Genetic Algorithms

The system described in this article further processes the results of a stemmer that has been developed for highly inflectional languages [3]. This stemmer divides each word in two parts, a stem and a suffix. The stemmer has been based on the concept of genetic algorithms [9] [10], by defining individuals that when combined with the original list of words, produce the solution proposed. These individuals are arrays of integers, each integer depicting a segmentation boundary of a specific word, i.e. the number of letters that comprise the stem [11]. The remainder of the word represents the suffix.

The objective function that has been experimentally found to give the best results utilizes a set of training data, comprising correctly segmented words. The fitness of each individual is determined by comparing the frequencies of appearance of the suffixes in the training set with those in the solution proposed by that individual. The higher the resemblance between the two sets, the higher the fitness of the specific individual. The training set comprises a limited number of words that are arbitrarily selected and provide a prototype according to which all other words will be processed. The reason that this function is effective is that macroscopically the frequency of appearance of a certain suffix in a given set of words remains to a large degree unaltered, irrespectively of the corpus chosen.

A novel approach was adopted, to address the problem of high-dimensionality, that occurs when large corpora are examined, according to which each individual is segmented into smaller equally sized parts. Each of these parts evolves through a number of generations independently. Thereby, instead of one high-dimensional individual the GA is "decomposed" to processing many smaller ones. These smaller subsets are combined at regular intervals, every μ iterations, updating the values for all the elements of the individual. Next the individuals are randomly disassembled once more in subsets different than the original ones, and this procedure is repeated until the GA meets a termination criterion.

One of the main advantages of the GA-based stemmer lies in its ability to learn by example using only limited language-specific knowledge. Thereby it is easily adaptable to different languages. Moreover, by setting a standard according to which the system will perform the segmentation via the training set, the system becomes easily customizable by a linguist without serious alterations to its main structure.

III. Hierarchical Clustering

The main reason for selecting a hierarchical instead of a partitional clustering algorithm is the lack of apriori knowledge about the specific number of clusters in the ideal partition. Moreover, the nested structure of hierarchical clustering reveals connections and interdependencies between different clusters, a feature that is desirable when examining natural languages. For example a verb in passive voice and a verb in active voice belong to different clusters (inflectional paradigms), they are however relevant since they both belong to the general category of verbs.

There are two general approaches regarding hierarchical clustering. The most commonly used is agglomerative clustering, where each data object is initially assigned to its

own singleton cluster. At each step, the closest pair of clusters is merged, until all objects are combined in an all-inclusive cluster. The second approach is called divisive. It starts off with an all-inclusive cluster and progressively splits one cluster at each step, until only singleton clusters remain. In both approaches, a notion of cluster proximity must be defined in order to decide which clusters should be merged or split, respectively. Depending on the metric adopted to calculate distances, different clustering methods, and thus results, are defined.

In this paper agglomerative clustering methods will be used. In the initial step, where only singleton clusters exist, cluster proximity depends solely on the appropriate distance between the vectors of the objects' representations. Moreover, when calculating the proximity of clusters comprising more than one object, there are several alternatives. The complete-link or MAX agglomerative clustering assumes that the distance of a pair of clusters is the maximum pairwise distance between the objects of the two clusters. Likewise, in the single-link or MIN alternative the distance between two clusters is defined as the minimum pairwise distance between the objects of the two clusters. Finally, in the average-link version, cluster proximity is the average pairwise distance between all objects of the clusters. There is also the alternative of representing all data objects of each cluster with a centroid, and whenever cluster proximity needs to be calculated it refers to the distance of the two centroids.

IV. Clustering Algorithm Description

Agglomerative clustering initially assigns each data object to its own cluster and proceeds by merging the pair of clusters with the smallest pairwise distance until all clusters are merged. In order to efficiently process a large amount of data, the algorithm proposed here is allowed to merge more than one pair of clusters in each step, if the distances between them are equal to one another and equal to the minimum distance between any clusters. The only constraints involve preventing clusters that have already participated in a group, from being merged again into more groups in the same step. For example, if cluster pairs (C_A, C_B) and (C_B, C_C) have the same smallest distance, the algorithm merges only one of them, i.e. the first that is recognized, leaving the other unchanged. That's because once the two clusters (e.g. C_A and C_B) have been merged they are perceived as one united cluster and therefore all the other distances that have been calculated prior to the merge are no longer valid. The major steps of the algorithm implemented are presented in Figure 2.

A. Data Objects

The clustering system makes use of a list of words that are segmented into stems and suffixes. As mentioned in the introduction, the main goal is to recognize morphological analogies between pairs of stems and to group each such pair into the same cluster, the characteristics that will be used to identify those analogies are the distinct suffixes that are linked to each stem. Using the suffixes as characteristics, every stem is a single data object and its representation is a vector of binary values, the dimension of which equals the number of different suffixes that have been identified in the corpus. If a stem is linked to a certain suffix then the value of the corresponding element in its vector becomes "1", otherwise it becomes "0".

1. Assign each data point to its own singleton.
2. **repeat**

→ Determine the minimum pairwise distance between clusters

→ Create a group containing the pairs of clusters with the minimum distance.

→ Merge each of those pairs unless at least one of the clusters in it has already been merged in the same step.

3. **until** only one cluster remains

Figure 2. Steps of the agglomerative clustering

One of the main difficulties facing this representation is that, although the value "1" clearly states that the corresponding suffix is part of the stem's inflectional paradigm, a value of "0" cannot reject such a claim. On the contrary, a value of "0" may suggest either one of two possible events:

- The corresponding suffix doesn't belong to the stem's inflectional paradigm, or
- The corresponding suffix belongs to the stem's inflectional paradigm but the specific word-form wasn't included in the corpus being studied. In this case the "0" value will be hereafter mentioned as a "hidden 1".

This duality of the value "0" may necessitate modifications to the clustering approach selected, and more specifically to the distance metric, as shall be detailed in the next section. The second limitation is that due to the binary representation, a centroid has no actual meaning i.e. a suffix is either present or not present in an inflectional paradigm. Moreover, using a medoid to represent the objects of a cluster is also questionable, because elements with a value of "0" are not as significant as the elements with value "1". Therefore, when calculating cluster proximity, the experiments will be restricted to MIN, MAX and average distances.

B. Distance Between Data Objects

Whenever the algorithm needs to determine the proximity of two clusters, it calculates the distances between all possible pairs of their objects.

The first distance examined in this paper is City Block which in the case of binary data actually reduces to the *Hamming* distance. The *Hamming* distance corresponds to the number of elements for which the binary strings examined are different. Though this distance can inform about whether two data objects (stems, in the present application) are similar, it may lead to confusing results when it comes to determining the distance between two vectors with a large degree of dissimilarity. In the latter case, the absence of a suffix from one stem (denoted by "0" in the corresponding element of its vector) that is present in the other stem's vector, does not necessarily indicate that the two stems belong to different

inflectional paradigms. The absence of a suffix might be attributed to the fact that the corpus is not very extensive, and therefore does not include the word that is formed by the concatenation of the given stem and suffix, even though such a word actually exists.

To overcome this problem, the system attempts to extract information from the data available, and utilizes this information to generate a distance that will better suit the specific data objects. This new distance, hereafter denoted as *Morph_Stat* distance, should still contribute a zero term when the corresponding elements have the same value among the two vectors being examined, but in the case of different values it must produce a term in the range of (0,1) instead of exactly 1, which is what the *Hamming* distance assumes. The value of the term will reflect the likelihood that the zero value in a stem's vector might actually be a "hidden 1". The closer the term is to 0, the higher the possibility of an actual "hidden 1". A mathematical formula for calculating the likelihood of a "hidden 1" is provided within this subsection.

To clarify the above assumption, a numerical example is provided in Figure 3, presenting two vectors of 10 elements (corresponding to two stems with a total of 10 suffixes) and the distance between them as calculated by both the *Hamming* and the *Morph_Stat* distances. Figure 3 shows that, in contrast to the *Hamming* distance, which increases the total value by one each time the two vectors differ in an element value, *Morph_Stat* increases it by a quantity $r_i \in [0,1]$ which represents the likelihood that the zero is actually a "hidden 1".

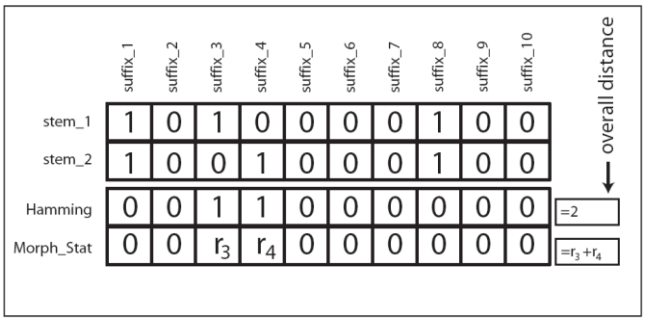

Figure 3. The results of two different distances (*Hamming* and *Morph_Stat*) between data objects.

In the example of Figure 3, suffix_3 is present in stem_1 but not in stem_2. In this case, *Morph_Stat* needs to determine whether this absence is attributable to different inflectional paradigms, or to the specific corpus that didn't include the existing word-form (i.e. a "hidden 1" event). The following equation is used to calculate r_i:

$$r_i = 1 - \frac{\sum_{k \in STEM} P(suf_i \mid suf_k)}{\sum_{l=1, l \neq i}^{N} P(suf_i \mid suf_l)} \quad (1)$$

In the nominator, the summation is made over the set *STEM*, which comprises all suffixes that are present in the vector from which suffix i is missing (suffixes suffix_1, suffix_4 and suffix_8 for the case of r_3 in the example presented in Figure 3). Each of the terms $P(suf_i \mid suf_k)$ of the sum expresses the probability of appearance of suffix i given suffix k. The denominator expresses the same sum for all suffixes, whether their corresponding values are "0" or "1", except for suffix i. The denominator remains the same for any given suffix i; as a result the value of the fraction depends entirely on the nominator. In the example of Figure 3, r_3 is calculated as follows:

$$r_3 = 1 - \frac{P(suf_3 \mid suf_1) + P(suf_3 \mid suf_4) + P(suf_3 \mid suf_8)}{\sum_{l=1, l \neq 3}^{10} P(suf_3 \mid suf_l)} \quad (2)$$

A large value of the nominator is translated into a high probability that the value zero in the stem's vector is attributed to the absence of the word from the corpus and thus corresponds to a "*hidden 1*", rather than to a different inflectional example. This assumption agrees with the fact that the fraction will tend to one and thus the value of the distance element r_i will tend to zero. In that case, even if the corresponding elements are different between the vectors compared, the overall distance between them will increase only by a small amount because of the "*hidden 1*".

The conditional probabilities for the specific example of Figure 3 have been calculated based on the experimental data and are shown in Table 2.

Table 2. Conditional probabilities $P(suf_i \mid suf_l)$ of appearance of one stem suf_i given the appearance of another suf_l.

$P(suf_i \mid suf_l)$	***i*=3**	***i*=4**
***l*=1**	0.001	0.001
***l*=2**	0.002	0
***l*=3**	1	0
***l*=4**	0	1
***l*=5**	0	0
***l*=6**	0	0
***l*=7**	0	0
***l*=8**	0.001	0.001
***l*=9**	0	0
***l*=10**	0	0

Using the conditional probabilities of Table 2, r_3 becomes equal to 0.5 while r_4 becomes equal to 1. The results indicate that according to the data there is a 50% probability that the zero value in the third element of stem_2 might be a *"hidden 1"*. Therefore the overall distance is 1.5, which is slightly smaller than the corresponding *Hamming* distance.

V. Experiments

The experiments presented in the following subsections examine which of the main approaches for cluster proximity (MAX, MIN or average) is the most appropriate for the given data. Moreover, the two distance metrics (Hamming and Morph_Stat) are checked for a range of different sizes of data to determine which metric is the most suitable. To minimize the effects of the errors of the GA stemmer while evaluating

the effectiveness of the clustering schemes, the correct segmentation of the words, according to the morphological lexicon [4], is used instead of the actual outcome of the GA stemmer [3]. Moreover the morphological lexicon is also used as a reference when validating clustering results, providing the inflectional paradigm of each stem and thus its class label.

A large set of 213,000 distinct words was used to acquire the data objects (i.e. the stems) needed for experimentation. These 213,000 words were extracted from various corpora by inserting each distinct word only once at the point of its first appearance. This set of words contains a total of 26,600 distinct stems.

Before presenting the experimental results, it would be useful to determine a measure of cluster validity that will allow comparisons between different experimental configurations. The agglomerative algorithm has been developed in C++ and the experiments have been executed on a PC with a single Intel Pentium processor operating at a frequency of 3.4 GHz.

A. Cluster Validation

A number of supervised measures can be used for evaluating the clustering results, since there exist external information from the manually-created morphological lexicon, in the form of class labels for each of the data objects (stems). The approach chosen for the experiments measures the extent to which two objects that have been assigned to the same cluster also belong to the same class and vice versa. The validity measures examined require the computation of the following four quantities:

- f_{00}, which corresponds to the number of pairs of objects that have been assigned to different clusters and indeed belong to different classes (indicating a correct clustering).
- f_{10}, which corresponds to the number of pairs of objects that have been assigned to different clusters even though they belong to the same class (indicating an incorrect clustering).
- f_{01}, which corresponds to the number of pairs of objects that have been assigned to the same cluster even though they belong to different classes (indicating an incorrect clustering).
- f_{11}, which corresponds to the number of pairs of objects that have been assigned to the same cluster and indeed belong to the same class (indicating a correct clustering).

The most frequently-used validity measures that are based on these four quantities are the *Rand statistic* [12] and the *Jaccard* coefficient [13]. The mathematical expressions of the two measures are provided in equations (3) and (4) respectively:

$$Rand\_statistic = \frac{f_{00} + f_{11}}{f_{00} + f_{10} + f_{01} + f_{11}} \quad (3)$$

$$Jaccard\_coefficient = \frac{f_{11}}{f_{00} + f_{10} + f_{01} + f_{11}} \quad (4)$$

In both cases, the nominator is used to enumerate correct assignments of data objects into clusters and the denominator to enumerate both correct and incorrect assignments. The larger the value of equation (3) or (4), the more successful the corresponding clustering is. The results of an agglomerative clustering using the *Hamming* distance and the average-link approach are illustrated in Figure 4.

As can be seen, both the *Rand* and the *Jaccard* metrics improve initially, as the number of clusters is reduced (e.g. from 180 to 60 clusters). For a relative wide range (between 90 and 40 clusters) this remains virtually unchanged. Only when the number of clusters is reduced further (to less than 30 clusters) do both cluster validity measures start to fall. Furthermore, both measures peak at approximately the same value, though the peak of the *Jaccard* coefficient is more marked.

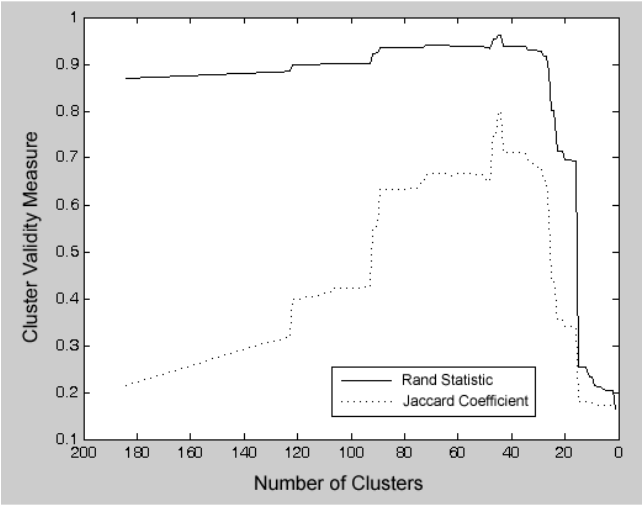

Figure 4. Evaluation of a clustering algorithm for 1,000 stems using the *Rand* statistic and the *Jaccard* coefficient.

These results refer to a set of 1,000 stems (data objects) where the algorithm has been evaluated according to both validity measures, to examine the ability of each validity measure to distinguish between different clusterings for the specific data.

The main difference between the two validity measures is that in the case of the *Jaccard* coefficient the cases calculated in f_{00} are not taken into account in either the nominator or the denominator. Since these are the most numerous instances (especially in a multi-class case such as the one examined in this paper), their removal from the calculation essentially removes a practically constant component and allows a more accurate definition of the clustering results. Therefore, even though the *Jaccard* coefficient leads to smaller absolute values, it depicts more clearly the peak of the algorithm.

For the remaining examples presented in this paper, cluster validity is determined via the *Jaccard* coefficient to avoid any excess noise imposed by the f_{00} cases.

B. Cluster Proximity

Three different variants, namely the MAX or complete-link (Eq. (5)), the MIN or single-link (Eq. (6)) and the average-link (Eq. (7)), are studied to determine which cluster proximity approach is better suited to the specific application. The three alternatives are examined for both distance functions on a set of 1,000 data objects (stems) and the results are illustrated in Figure 5.

$$d_{\max}(C_i, C_j) = \max_{a \in C_i, b \in C_j} [d(a,b)] \quad (5)$$

$$d_{\min}(C_i, C_j) = \min_{a \in C_i, b \in C_j} [d(a,b)] \quad (6)$$

$$d_{average}(C_i, C_j) = \frac{\sum_{a \in C_i, b \in C_j} d(a,b)}{n_i \cdot n_j} \quad (7)$$

where n_i and n_j are the number of elements comprising clusters C_i and C_j correspondingly and $d(a,b)$ is the distance between two vectors, one selected from each cluster at a time.

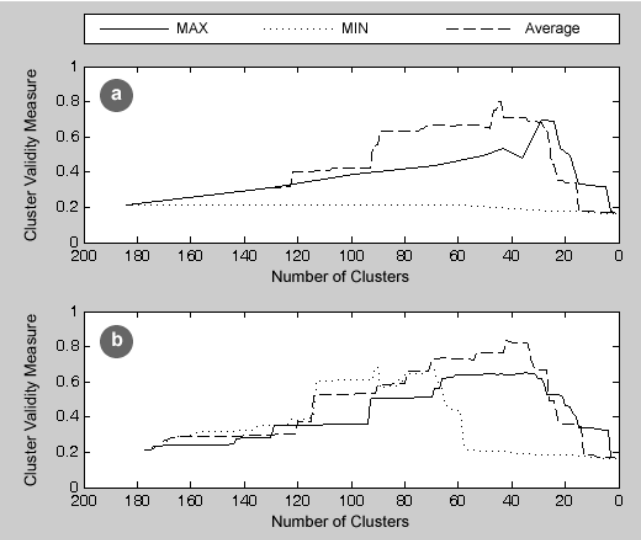

Figure 5. Three concepts of cluster proximity using a) the *Hamming* distance and b) the *Morph_Stat* distance.

The results of Figure 5 demonstrate that for both the *Hamming* distance (Fig. 5a) and the *Morph_Stat* distance (Fig. 5b), average-link is the variant that leads to the highest clustering results. The explanation for the superiority of the average-link variant lies in the fact that the data objects are incomplete and therefore a single data object cannot be indicative of the whole cluster. In most cases, only a small portion of the available word-forms for a given stem appears in the examined corpus, and therefore the corresponding vector is not fully descriptive. In contrast, by applying average-link the missing information for each single data object is balanced to a certain degree by extracting relevant information regarding the cluster from all objects within the cluster.

It is clear from Figure 5 that the *Morph_Stat* distance is better suited than the *Hamming* distance for the task at hand. In the case of MIN (single-link), the *Hamming* distance fails to find even a single better clustering than the initial singletons. This is attributable to the fact that the minimum distances are usually recorded between vectors with few "1" elements which as a rule correspond to stems that cannot provide credible information.

C. Varying the Corpus Size

The next series of experiments examines the effect of the number of stems that are processed by the clustering algorithm. For this reason, different numbers of stems (data objects) are selected for processing. Each stem is inserted in the list at the point of its first appearance in the corpus. Therefore, stems near the beginning are as a rule more frequent than stems near the end of the list.

Five different sets of stems are examined, comprising from 1,000 to 5,000 stems. Each set contains all previous stems plus an additional 1,000 that are next in line in the list of all stems. The results, which are obtained utilizing the average-link cluster proximity, are depicted in Table 3 where the validity of the best clustering is presented along with the number of clusters formed.

Table 3. Clustering results for various numbers of stems

		Number of Stems				
		1,000	**2,000**	**3,000**	**4,000**	**5,000**
	Actual classes	57	76	91	96	102
Hamming	**Validity**	0.799	0.724	0.707	0.725	0.733
	Clusters	44	50	82	100	86
	Steps	74	126	150	170	225
Morph_Stat	**Validity**	0.834	0.774	0.763	0.779	0.754
	Clusters	42	65	61	83	96
	Steps	137	247	351	427	505

The results in Table 3 indicate that the *Morph_Stat* distance is superior to the *Hamming* one for the given problem, regardless of the number of stems processed. The fact that the number of clusters formed in the optimal case is smaller than the actual classes is attributed to the fact that the vectors of the stems don't include all the information needed to make a correct decision, i.e. all the suffixes linked to the stem in the morphological lexicon. Thus, the ability to differentiate between the actual classes is reduced. A typical example is a case where two stems belong to inflectional paradigms that differ only by one suffix. If the word-forms created by the concatenation of these stems and the corresponding suffix are not included in the corpus, then the algorithm will not be able to distinguish between the two and will thus group them in the same cluster.

Moreover, when employing the *Morph_Stat* distance, the algorithm requires more steps to achieve the optimal clustering although the number of clusters resulting at this optimum is approximately the same. This observation reveals that in the case of *Morph_Stat* the algorithm becomes more "cautious" in terms of avoiding the grouping of multiple clusters at each step. The *Morph_Stat* distance provides a representation of cluster proximity characterized by a higher level of detail.

The fact that the algorithm is sensitive to the information provided by the stems is illustrated by the following experiment. Two different sets comprising 1,000 stems each are examined; one using the first 1,000 stems encountered in the corpus and the other with stems 4,001 to 5,000. The second set contains less frequent stems, and therefore, in general, fewer word-forms corresponding to each of those stems are

present in the corpus. The results are illustrated in Figure 6.

Although the number of stems is the same, clustering of the second set leads to less accurate results than for the first set when compared to the manually crafted morphological lexicon. This is due to the fact that the information provided by those stems is less thorough. More specifically the first set of 1,000 stems (set a) is extracted from 6,358 words, resulting in an average number of 6.4 suffixes per stem while in the second set (set b) the corresponding number of words is 5,390 (5.4 in average).

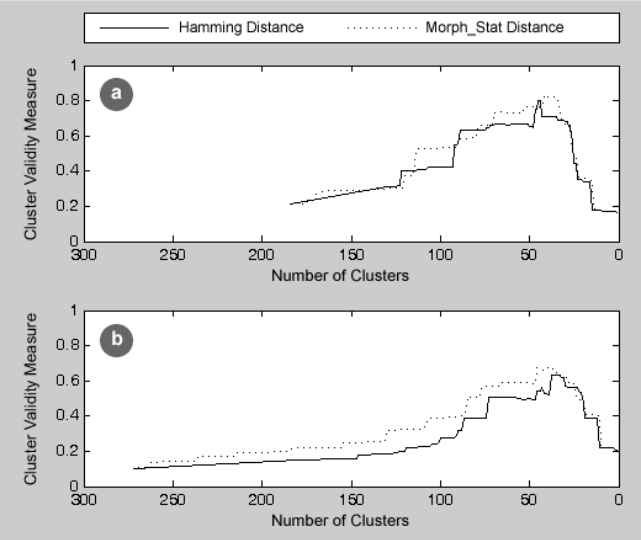

Figure 6. Agglomerative clustering for two different sets of stems : a) stems 1-1,000 and b) stems 4,001-5,000.

Apart from the average number of suffixes per stem, one should also examine how they are distributed among the actual number of suffixes. In the experiments presented in this article, words are distinguished along three different grammatical categories, namely: verbs, nouns and adjectives. Each of these categories has a different number of suffixes assigned to their inflectional paradigm as shown in Table 4.

Table 4. Average number of suffixes per grammatical category

	Verbs	**Nouns**	**Adjectives**
Average number of suffixes / inflectional paradigm	11.7	4.3	7.8

Figure 7 illustrates the distribution of the number of stems per suffix for the two datasets examined in this specific experiment.

Three peaks are prominent in both histograms of the two datasets. The first peak is around the value "4" suffixes per stem and mostly corresponds to inflectional paradigms of nouns. The second peak is around "6" and the third is around "10" referring to adjectives and verbs correspondingly.

The difference between the two histograms is that the range around those center peaks is narrower on the first set of 1,000 stems while in the second set the range appears wider. Moreover the silhouette surrounding the histogram of the first set tends to shift to the right and thus to more suffixes per stem, while in the second set that shift appears to point in the opposite direction.

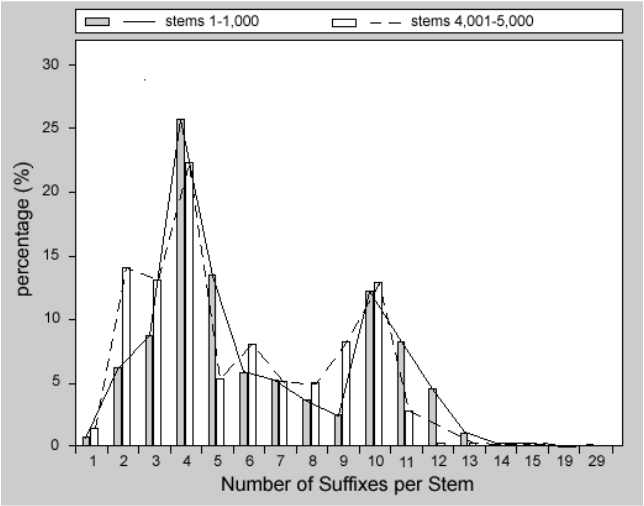

Figure 7. Distribution of the number of suffixes per stem for two data sets : a) stems 1-1,000 and b) stems 4,001-5,000.

These observations further support the original assumption that the second set, comprising rarer stems, contains less information and thus unavoidably leads to less accurate clustering results for the system.

D. Validity among Grammatical Categories

To further investigate the system performance, the results are examined according to the three grammatical categories mentioned above: verbs, nouns and adjectives. Three different sets were created for the three corresponding categories, comprising 1,000 distinct stems each. Table 5 depicts the best clustering accuracy achieved as well as the number of ideal classes and of the clusters that correspond to the best clustering. The metric used in this series of experiments is the *Morph_Stat* distance, since it leads to better clustering.

Table 5. Clustering results per grammatical category

	Verbs	**Nouns**	**Adjectives**
Actual Classes	57	35	9
Validity of the Best Clustering	83.5	73.5	82.1
Number of Clusters for the Best Clustering	42	26	21

The results of Table 5 illustrate that the most problematic of the three grammatical categories is that of nouns. The lower accuracy level of this specific category is mainly attributable to the fact that the inflectional paradigms of nouns contain fewer suffixes in Modern Greek and are therefore harder to distinguish. In fact there is a number of cases where different inflectional paradigms are assigned the same set of suffixes as in the example presented in Table 6.

The stems presented in Table 6 are assigned to the same cluster early on in the clustering process, since the distance between them is zero according to both metrics (*Hamming* and *Morph_Stat*). The inflectional paradigms these two stems belong to are obviously different as the first one, "αλλαγή", is a noun in feminine gender while the second one, "ιδιώτης", is a

noun in masculine gender. It should be pointed out that the word form corresponding to nominative singular, for "αλλαγή" has the same suffix as the word form corresponding to genitive singular for "ιδιώτης".

Table 6. Example of different inflectional paradigms containing the same set of suffixes

Stem	English translation	Inflectional Suffixes
αλλαγ-	change	-η, -ης, -η, -η, -ες, -ων, -ες, -ες
ιδιώτ-	private	-ης, -η, -η, -η, -ες, -ων, -ες, -ες

Contrary to what is observed in nouns, verbs and adjectives have a greater number of suffixes assigned to their inflectional paradigms (as reflected in Table 4), making it easier to differentiate between the classes. Another interesting remark on the results of Table 5 is that even though data from the two grammatical categories belong to different numbers of classes (9 classes for the adjectives and 57 for verbs for the given datasets), the clustering accuracy is virtually the same. This observation indicates that in the implementation under study, clustering results are somewhat independent of the actual number of classes.

VI. System Modification

The way the clustering algorithm is implemented prohibits the simultaneous processing of large datasets. For instance, the system requires up to 10 days to process 10,000 stems.

By thoroughly examining the dataset we come to the conclusion that many data objects have identical vectors. This was actually expected since there are a limited number of inflectional paradigms. Even though data objects with identical vectors are immediately grouped in the same cluster (since they have zero distance), they still increase the computational complexity, since the system utilizes the average-link approach to calculate distances between clusters. To overcome this problem and thereby decrease the execution times, a new approach was created in which each vector is selected once, forming a set of unique vectors that are thereafter clustered. Each unique vector is assigned to more than one stems, which in turn belong to the cluster the corresponding vector has been attributed to. The decrease in the dimensionality is even more profound for large data sets as illustrated in Table 7.

Table 7. Number of unique vectors for various data sets

Number of stems	1,000	2,000	3,000	4,000	5,000
Number of unique vectors	184	316	420	521	613

To maintain the information provided by the number of stems assigned to a single vector, each vector participates in the distance calculation with a different weight, corresponding to the number of stems assigned to it. With this new approach, equation (7) is transformed to equation (8).

$$d(C_i, C_j) = \frac{\sum_{a \in C_i, b \in C_j} d(a,b) \cdot n_a \cdot n_b}{\sum_{a \in C_i, b \in C_j} n_a \cdot n_b} \quad (8)$$

where a and b are the unique vectors of clusters C_i and C_j correspondingly and n_a and n_b are the number of stems attributed to each of those vectors.

The clustering accuracy achieved by the implementation on this new approach is identical to the previous one while the execution times have been reduced substantially. Experimental results indicate that the reduction is up to 82% over the original execution times.

VII. Conclusions

In this article, the application of hierarchical clustering to grouping stems with similar characteristics has been reported. Experimental data indicate that agglomerative clustering can be successfully applied to the task of grouping stems in inflectional paradigms. Moreover, the *Morph_Stat* distance which utilises statistical information from all available stems manages to overcome, to a certain degree, the limitations caused by insufficient data and consistently outperforms the *Hamming* distance. Future work focuses on identifying automatically a threshold value for the distance for which the best clustering occurs so as to terminate the algorithm at that specific point. It is expected that this line of work can lead to a substantial reduction in the human effort needed to create a morphological lexicon in Modern Greek.

Acknowledgments

The authors would like to thank Ms. M. Vassiliou of the ILSP for her valuable help providing insights on the study & characteristics of the Greek language. This research has been supported by the PENED programme 03ED97, funded by the Greek Secretariat for Research and Technology.

References

[1] Petasis, G., Karkaletsis, V., Farmakiotou, D., Samaritakis, G., Androutsopoulos, I., Spyropoulos, C. "A Greek Morphological Lexicon and its Exploitation by a Greek Controlled Language Checker". In *Proceedings of the 8th Panhellenic Conference on Informatics (PCI 2001)*, Nicosia, Cyprus, 8-10 November, pp. 80-89.

[2] Detorakis, Z., Tambouratzis, G. "Implementation of a Multi-Objective Genetic Algorithm on Word Segmentation in Modern Greek". *In Proceedings of the 11th IASTED International Conference on Artificial Intelligence and Soft Computing (ASC 2007)*, Mallorca, Spain, 29-31 August 2007.

[3] Detorakis, Z., Tambouratzis, G. "Introduction of a Sectioned Genetic Algorithm for Large Scale Problems". In *Proceedings of the 2nd International Conference on Bio-Inspired Models of Network, Information, and*

Computing Systems (Bionetics 2007), Budapest, Hungary, 9-13 December 2007.

[4] Gavrilidou, M. "The ILSP morphological lexicon and morpho-syntactic tagger". *Internal report*. Athens: Institute for Language & Speech Processing (in Greek), 1996.

[5] Goldsmith, J. "Unsupervised Learning of the Morphology of a Natural Language". *Computational Linguistics*, 27 (2), pp. 153-193, 2001.

[6] Jain, A., Murty, M., Flynn, P. "Data clustering: a review. ACM", *Computing Surveys*, 31 (3), pp. 264-323, 1999.

[7] Duda, R.O., Hart, P.O., Stork, D.G. *Pattern Classification (2nd Edition)*, Wiley-Interscience, 2000

[8] Kaufman, L., Russeeuw, P. *Finding groups in data: an introduction to cluster analysis*. John Wiley and Sons, 1990.

[9] Goldberg, D.E. *Genetic algorithms in search, optimisation and machine learning*, Addison-Wesley, Boston, 1989.

[10] Uysal, O., Bulkan S. "Comparison of Genetic Algorithm and Particle Swarm Optimization for Bicriteria Permutation Flowshop Scheduling Problem", *International Journal of Computational Intelligence Research*, 4(2), pp. 159–175, 2008.

[11] Kazakov, D., Manandhar, S. "A hybrid approach to word segmentation". In *Proceedings of the 8th International Workshop on Inductive Logic Programming (ILP-98)*, Madison, Wisconsin, USA (July 22-24), Springer, 125-134, 1998.

[12] Hubert, L., Arabie, P. "Comparing partitions", *Journal of Classification*, 2 (1), pp. 193-198, 1985.

[13] Halkidi, M., Batistakis, Y., Vazirgiannis, M. "Cluster validity methods: part I", *ACM SIGMOD Record*, 21 (2) pp.40-45, 2002.

Author Biographies

Zacharias Detorakis obtained his Diploma in Electrical Engineering and Computer Science and his Ph.D. degree, both from the National Technical University of Athens (NTUA) in 2003 and 2009 respectively. He has been working at the Institute for Language and Speech Processing in Athens since 2005. His research interests include evolutionary computation and natural language processing applications.

George Tambouratzis received the Diploma in electrical engineering from the National Technical University of Athens Greece, in 1989, and the M.Sc. degree in digital systems and the Ph.D. degree in neural networks and pattern recognition, both from the Department of Electrical Engineering, Brunel University, U.K., in 1990 and 1993, respectively. Since 1996, he has been associated with the Institute for Language and Speech Processing, Athens, Greece, where he currently holds a research post. His research interests include neural networks, pattern recognition, evolutionary computation and computational linguistics. He is a member of the Technical Chamber of Greece and the IEEE.