Stemming for Kurdish Information Retrieval

Shahin Salavati[1], Kyumars Sheykh Esmaili[2], and Fardin Akhlaghian[3]

[1] University of Kurdistan
Sanandaj, Iran
shahin.salavati@ieee.org
[2] Nanyang Technological University
Singapore
kyumarss@ntu.edu.sg
[3] University of Kurdistan
Sanandaj, Iran
f.akhlaghian@uok.ac.ir

Abstract. Resource scarcity along with diversity –in both dialect and script– are the two primary challenges in Kurdish language processing. In this paper we aim at addressing these two problems by building stemmers for the two main dialects of the Kurdish language (i.e. Sorani and Kurmanji) and investigate their effectiveness on Kurdish Information Retrieval.

More specifically, we build *Jedar*, the first rule-based stemmer for both Sorani and Kurmanji. We also implement *GRAS* –as a state-of-the-art statistical stemming technique– and apply it to both of the Kurdish dialects. We then conduct a comprehensive experimental study to compare the effectiveness of these stemmers.

Our experimental results show that stemming can significantly –up to %35– improve the retrieval performance on Kurdish documents. Furthermore, they indicate that the gains from the rule-based and the statistical approaches are comparable.

1 Introduction

Stemming is a common form of language processing in most information retrieval (IR) systems. Stemming is the process of reducing a word to its stem or root form. It allows documents in which a term is expressed using a different morphological form from the query, to be found and matched.

Although experiments with English data show mixed results [9,11,14], retrieval performance for morphologically more complex languages (e.g., Hungarian, Czech and Bulgarian [17], German [20,3], Dutch and Italian [20], and Arabic [33]) has benefited consistently and significantly from stemming.

The Kurdish language is an Indo-European language spoken in Turkey, Iran, Iraq and Syria. Despite having a large number of speakers, Kurdish is considered a less-resourced language for which –among other basic tools– no stemmer has been developed. Apart from the resource-scarcity problem, diversity –in both dialect and writing systems– is another primary challenge in Kurdish language processing. In fact, Kurdish

R.E. Banchs et al. (Eds.): AIRS 2013, LNCS 8281, pp. 272–283, 2013.

is considered a *bi-standard* language [7,10]: the Sorani dialect written in an Arabic-based alphabet and the Kurmanji dialect written in a Latin-based alphabet. The features distinguishing these two dialects are phonological, lexical, and morphological.

This paper reports on our efforts in building stemmers for the two main dialects of the Kurdish language and investigate their effectiveness on Kurdish IR. The main contributions of this work are:

- we build *Jedar*, a rule-based stemmer for both Sorani Kurdish and Kurmanji Kurdish,
- we implement *GRAS* –a state-of-the-art statistical stemming technique– and apply it to both of the Kurdish dialects,
- we conduct a comprehensive experimental study to compare the effectiveness of these stemmers (including sensitivity analysis to fine-tune their parameters), and
- we carry out a detailed analysis of the results to obtain insights about the behavior of each configuration.

Additionally, our source codes for the Jedar and GRAS implementations along with the list of Kurmanji and Sorani suffixes used in our experiments are freely accessible and can be obtained from [12]. We hope that making these resources publicly available, would bolster further research on Kurdish IR.

The rest of the paper is organized as follows. We first, in Section 2, give a little bit of background on stemming in IR and also on the Kurdish language and dialects. Then in Section 3 we present the Kurdish suffixes and show how we used them to build Jedar, our rule-based stemmer. In Section 4, we briefly explain the GRAS statistical stemming algorithm [22] as well as our implementation of this algorithm. The details of our experimental study and analysis are reported in Section 5. Finally, we conclude the paper in Section 6.

2 Background

In this section we first give an overview of stemming in IR and then briefly introduce the Kurdish language and dialects.

2.1 Stemming for Information Retrieval

In an IR system, stemming is used to reduce variant word forms to common roots, and thereby improve the ability of the system to match query and document vocabulary. The variety in word forms comes from both inflectional and derivational morphology [32]. Inflection characterizes the changes in word form that accompany case, gender, number, tense, person, mood, or voice. Derivational analysis reduces surface forms to the base form from which they were derived, and includes changes in the part of speech [11].

All stemming algorithms can be roughly classified as rule-based (a.k.a affix removing) or statistical. Below we give a brief overview of each of these classes.

Rule-Based Stemmers. Rule-based stemmers apply a set of transformation rules to each word, trying to strip its suffixes[1]. Two of the most popular algorithms in English IR, the Lovins stemmer and the Porter stemmer, are based on suffix removal.

Lovins' paper [15] was the first published description of a stemmer. It defines 294 endings, each linked to one of the 29 conditions, and the 35 transformation rules. For a word being stemmed, an ending with a satisfying condition is found and removed. The algorithm is fast but misses certain endings.

Porter's algorithm [23] defines five successively applied steps of word transformation. Each step consists of set of rules. The algorithm is concise (about 60 rules) and efficient. The main flaws and errors are well-known and can mostly be corrected with a dictionary. The idea of Porter algorithm was later generalized into a stemmers framework called Snowball [24].

The major drawback of the rule-based approach is its dependency on a priory knowledge of the concerned language's morphology.

Statistical Stemmers. In contrast to the rule-based stemmers, statistical stemmers are language-independent and only require a corpus or a lexicon. In following we briefly summarize three important statistical stemmers.

The authors of the YASS stemmer [18] viewed stemming as a clustering problem in which the resulting clusters are considered as equivalence classes and their centroids as stems. Based on their implementation and experiments, they conclude that YASS' performance is comparable to rule-based stemmers like Porter or Lovins for English. For more morphologically-complex languages such as Bengali and French, YASS provides substantially improved performance as compared to using no stemming [18] .

Bacchin et al. [1] described a probabilistic model which relies on the mutual reinforcement relationship between stems and suffixes. Once the prefix and suffix scores are computed over a subset of documents from the corpus, the algorithm estimates the most probable split (into stem and suffix pair) for each word in the full corpus. A set of experiments with several languages produced equally good results as those produced by rule-based stemmers [1].

The main disadvantage of the aforementioned statistical stemming algorithms is that they are computationally expensive. In contrast, the recently-proposed GRAS algorithm [22] has been shown to be an efficient alternative. In experiments with seven languages of very different language families and varying morphological complexity, the authors showed that GRAS outperforms rule-based stemmers, three statistical methods (including YASS [18]), and the baseline strategy that did not use stemming.

Hence, we consider GRAS as the state-of-the-art solution for statistical stemming and use it in our experiments. We will describe the GRAS algorithm in more details later in Section 4.

2.2 The Kurdish Language and Dialects

Kurdish belongs to the Indo-Iranian family of Indo-European languages. Its closest better-known relative is Persian. Kurdish is spoken by 20 to 30 million people [8,10]

[1] Deletion of prefixes is not generally helpful for a stemming algorithm [11,21].

in Kurdistan, a large geographical area spanning the intersections of Turkey, Iran, Iraq, and Syria. It is one of the two official languages of Iraq and has a regional status in Iran.

Kurdish is a dialect-rich language, however, in this paper we focus on Sorani and Kurmanji which are the two closely-related and widely-spoken dialects of the Kurdish language [8]. Together, they account for more than 75% of native Kurdish speakers [31].

As summarized below, these two dialects differ not only in some linguistics aspects, but also in their writing systems.

Morphological Differences. Some of the important morphological differences are [16,8]:

- Kurmanji is more conservative in retaining both gender (feminine:masculine) and case opposition (absolute:oblique) for nouns and pronouns. Sorani has largely abandoned this system and uses the pronominal suffixes to take over the functions of the cases,
- the definiteness suffix `-aka` appears only in Sorani,
- in the past-tense transitive verbs, Kurmanji has full ergative alignment but Sorani, having lost the oblique pronouns, resorts to pronominal enclitics.

Scriptural Differences. Due to geopolitical reasons, each of the two dialects uses its own writing system: Sorani is almost-exclusively written in an Arabic-based alphabet and Kurmanji is almost-exclusively written in a Latin-based alphabet. Figure 1 shows the two standard alphabets and the mappings between them [5].

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Arabic-based	ا	ب	ج	چ	د	ێ	ف	گ	ژ	ک	ل	م	ن	ۆ	پ	ق	ر	س	ش	ت	وو	ڤ	خ	ز
Latin-based	A	B	C	Ç	D	Ê	F	G	J	K	L	M	N	O	P	Q	R	S	Ş	T	Û	V	X	Z

(a) One-to-One Mappings

	25	26	27	28
Arabic-based	ئ / ی	و	ی	ه
Latin-based	I	U / W	Y / Î	E / H

(b) One-to-Two Mappings

	29	30	31	32	33
Arabic-based	ڕ	ڵ	ع	غ	ح
Latin-based	(RR)	-	(E)	(X)	(H)

(c) One-to-Zero Mappings

Fig. 1. The Two Standard Kurdish Alphabets [5]

As we will explain in Section 3, these differences have direct implications on designing Kurdish stemmers.

3 Kurdish Stemming: Rule-Based Approach

In the following, we first present the main Kurdish suffixes and then introduce our rule-based suffix-removing stemmer.

3.1 Main Suffixes in Kurdish

Kurdish has a complex morphology [26,29,5] and one of the main driving factors behind this complexity is the wide use of inflectional and derivational suffixes [6]. In general, Sorani and Kurmanji share a large proportion of suffixes. However, there is a small, but very important, set of Sorani-specific suffixes. Below, we elaborate more on each of these two groups.

Suffix Group				Sorani	Kurmanji
Inflectional	Izafe Construction Markers	Masculine	Absolute	ی	(y)ê
			Oblique	ی	(y)î
		Feminine	Absolute	ی	(y)a
			Oblique	ی	(y)ê
	Plural Markers		Absolute	ان	(y)ên
			Oblique	ان	(y)an
	Definiteness Makers			یەک	ek
Derivational	Professional Nouns			وان	van
				کار	dar
				دار	kar
	Locational Nouns			خانە	xane
				ستان	stan
				گە	geh

(a) Noun Suffixes

Suffix Group		Sorani	Kurmanji
Personal Verb Endings & "To Be"	1st Person Singular	م	(i)m
	2nd Person Singular	ی	î
	3rd Person Singular	ێ/ا/ە	e
	1st Person Plural	ین	in/ne
	2nd Person Plural	ن	in/ne
	3rd Person Plural	ن	in/ne
Helper Verbs	Used in Infinitive Form	گرتن	girtin
		کردن	kirin
		بوون	bûn
		بردن	birin
	Used in Conjugated Form	بوو	bû
		کرد	kir
		کراو/کردوو	kirî/kiri
		دەکەن	dikin
		دەکات	dike

(b) Verb Suffixes

Fig. 2. Common Suffix Groups in Sorani Kurdish and Kurmanji Kurdish

Common Suffixes. An essential subset of common suffixes between Sorani and Kurmanji is depicted in Figure 2. The complete set can be downloaded from [12]. It should be noted that for some pairs, the Sorani and the Kurmanji strings are not complete transliteration-equivalents (based on the char-level mappings of Figure 1). The left side of Figure 2 (part a) contains the common *noun* suffixes. A few important remarks regarding this list are:

- the Izafe Construction is a shared feature of several Western Iranian languages [25]. It approximately corresponds to the English preposition *of* and is added between prepositions, nouns and adjectives in a phrase. The Kurmanji Izafe marker agrees in gender and in case with the head noun [30], thus giving rise to various forms,
- the impact of case in Kurmanji is also evident in the plural noun marker [30], for which two different forms exist,
- in the Kurmanji writing system, if suffixing results in two consecutive vowels, an extra *y* is inserted between them.

The right side of Figure 2 (part b) represents the common *verb* suffixes. There are two important notes here:

- in Sorani and Kurmanji while conjugating a verb in past or present tense, personal endings are added to the verb root. These endings –except in the past transitive tense– are identical to the present forms of the verb *to be* in Kurdish (بوون/ "bûn") [27].

– Kurdish resembles most Iranian languages in the fact that it possesses only a limited amount (around 300) of synthetic verbal lexemes [16,2]. Most verbal meanings in Kurdish are expressed through complex compositions. One important class of composition elements is auxiliary verbs which can be used in their infinitive form to build new nouns, or in conjugated form to build different tenses.

Sorani-Specific Suffixes. Compared to Kurmanji, Sorani has a richer set of suffixes. A small, but nonetheless very crucial, set of Sorani-specific suffixes is listed in Figure 3. As described below, the existence of these suffixes is due to Sorani's inherent

Suffix	Translit.	Description
م	*m*	1st Person Singular
ت	*t*	2nd Person Singular
ی	*i*	3rd Person Singular
مان	*maan*	1st Person Plural
تان	*taan*	2nd Person Plural
یان	*yaan*	3rd Person Plural

(a) MPM/Possessive Pronouns

Suffix	Translit.	Description
ەکە	*aka*	Definite Marker Singular
ەکان	*akaan*	Definite Maker Plural
دا	*daa*	A Common Postposition
ش	*sh*	A Common Conjunction

(b) Others

Fig. 3. Sorani-Specific Suffixes

morphological properties as well as its script and system of writing:

– Sorani uses the pronominal suffixes to take over the functions of the cases (see Section 2.2). The two principal uses of such suffixes are: (i) the mobile person markers (MPMs) [27,29] which are used as pronominal enclitics in past-tense transitive verbs (Kurmanji, in contrast, has full ergative alignment), and (ii) the possessive pronouns (Kurmanji, instead, uses the oblique form of the personal pronouns). As reflected in Figure 3a, these two suffix sets have identical representations.
– the definite markers (ەکە *aka* and ەکان *akaan* for singular and plural nouns, respectively) only exist in Sorani,
– there is a general tendency in Sorani's writing system to join suffixes to their preceding noun [5]. Its most prominent example is the verb بوون *boon* "to be" (presented in Figure 2b). Two other widely-used instances are the postposition دا *daa* –which is in fact the closing part of some commonly-used circumpositions and therefore has no independent meaning– and the conjunction یش *ish* "too".

3.2 Jedar

In this section we introduce **Jedar**[2], the first rule-based stemmer for the Kurdish language. Kurdish stems are often followed by multiple suffixes, hence, Jedar adopts a recursive approach to handle nested suffixes. Moreover, we have devised two techniques to decrease Jedar's over-stemming error[3]. The first technique –adopted from [15]– is to prevent over-stemming by setting a minimum stem length parameter, denoted by L.

[2] A Kurdish word (in Sorani: ژێدەر , in Kurmanji: *Jêder*) meaning "origin" in English.

[3] A common error in rule-based stemmers caused by blindly removing substrings that belong to the word's stem.

The second technique is to exploit the inherent suffixing properties of the Kurdish language. Below, we give a brief description of some of these properties:

• the nominal suffixes appear in a certain pre-defined order. To demonstrate this order, we analyze the example word کتێوەکانیشتاندا *ktewakaanishtaandaa* "[in] your books too" which consists of a stem (کتێو *ktew* "book") and four different suffixes:

دا	+	تان	+	یش	+	ەکان	+ کتێو	=	کتێوەکانیشتاندا
daa	+	*taan*	+	*ish*	+	*akaan*	+ *ktew*	=	*ktewakaanishtaandaa*
postpos.	+	poss. pron.	+	conjunc.	+	def. marker	+ stem	=	word

• in any given word, only one instance of each suffix type can appear. For example, although the word کلینیکەکە *klinekaka* "the clinic" contains both the indefinite marker (یەک *ek*) and the definite marker (ەکە *aka*), only the second one is a valid suffix and the first one should be left untouched. One important exception to this rule is MPMs/possessive pronouns (Figure 3a), which have identical representations but different roles.

• under some circumstances, the minimum length constraint can be relaxed. For example if a word ends in بوو *boo* "was", this string can be removed, as it is solely used to build the past perfect form of the verbs.

Jedar has been implemented as a single Java class (for both Sorani and Kurmanji) that takes a list of dialect-specific suffixes as input. For each input word, Jedar recursively removes the best matching suffix, taking into account a set of rules including those explained above.

4 Kurdish Stemming: Statistical Approach

As explained in Section 2.1, the GRAph-based Stemming (GRAS) algorithm has been shown to outperform a number of other existing statistical stemmers. Hence we chose this algorithm to compare with Jedar. The GRAS algorithm, in essence, consists of three steps [22]:

Step 1: Frequent Suffix Pair Identification. GRAS starts with a lexicon, a list of the distinct words of the concerned language (usually extracted from a corpus). The words in this lexicon are partitioned into a number of groups such that each pair of words drawn from a group has a common prefix of length at least λ, a pre-defined threshold. Within each group, all possible word pairs are enumerated and suffix pairs are extracted. For example, since the word pair $(w_1 = p||s_1, w_2 = p||s_2)$ share a common prefix p, then s_1 and s_2 constitute a candidate suffix pair. When all groups are exhausted, the total frequency of each suffix pair is computed and the non-frequent pairs (fewer occurrences than α, a cutoff threshold) are discarded.

Step 2: Graph Construction. Having built a list of frequent suffix pairs, this list is then used to construct a weighted undirected graph $G = (V, E)$ as follows. Each word in the lexicon is represented by a vertex in G. In this graph, the edge weight, $w(u, v)$, is the frequency of the suffix pair induced by the word pair represented by u and v. Needless to say, if $w(u, v) < \alpha$, there is no edge between u and v.

Step 3: Graph Decomposition. Once the graph G is constructed, the next step is to decompose it. The decomposition algorithm first chooses a pivotal node (say p) from the remaining vertices, such that its degree is maximized. Next, it considers the vertices adjacent to the pivotal node p one-by-one and measures the *cohesion* between p and v using the formula:

$$cohesion(p, v) = \frac{1 + |Adjacent(p) \cap Adjacent(v)|}{|Adjacent(v)|}$$

The value of cohesion lies between 0 and 1. If the cohesion value exceeds a certain threshold (δ), the vertex v is assumed to be morphologically related with the pivot p and is put in the same class as p. Otherwise, the edge (p, v) is deleted immediately to mark that p and v are not related.

As highlighted by the authors, the choices of the three main parameters –namely the minimum length of the common prefix λ, the suffix frequency cutoff α, and the cohesion threshold δ– are important for the performance of the algorithm. Although they provide some clues, but as shown later, more precise values can be found empirically.

The original implementation of GRAS is unfortunately not open source. Therefore, we built our own implementation from scratch by closely following the descriptions given in [22]. Our Java implementation of the GRAS algorithm (along with Jedar's implementation) can be obtained from [12].

5 Experiments

In our experiments, we used Pewan, a publicly-available Kurdish test collection [13] which contains two separate text corpora (one Sorani one Kurmanji) and a set of queries available in both dialects. The main properties of the Pewan collection are summarized in Table 1.

Table 1. The Pewan Test Collection [6]

	Number of Documents	*Number of Queries*	*Average QRel Length*
Sorani	115,340	22	42
Kurmanji	25,572	22	12.5

For the IR engine, we chose **MG4J** [19], an open-source Java retrieval system which has been shown [4,6] to be the best performing system for Kurdish IR among a number of systems.

In the following, we first report on the sensitivity analysis that we carried out to fine-tune Jedar's and GRAS' parameters. Then we provide more insights about the outcomes through a detailed analysis of the results.

5.1 Parameter Tuning

In these experiments we vary the stemming parameters and compare the results based on Mean Average Precision (MAP) values. Additionally, we also report the size of the resulting lexicon, that is the total number of distinct strings in Pewan after applying stemming.

Jedar's Parameter. We performed a sensitivity analysis on Jedar by varying the minimum stem length parameter, L, from 3 to 6 (according to Lovins [15], any useful stem often consists of at least three or four characters).

The results are shown in Table 2. In this table, *Baseline* denotes the case in which no stemming was applied. Based on these numbers, two important conclusions can be drawn: (i) our rule-based stemming solution generally improves the retrieval performance, (ii) for both Sorani and Kurmanji, the best result is achieved for $L = 3$ (the gains are 25% and 35% respectively).

Table 2. Tuning Jedar's Minimum Stem Length Parameter

Parameter	**Sorani**		**Kurmanji**	
Minimum Stem Length (L)	MAP	Lexicon Size	MAP	Lexicon Size
3	**0.440**	217522	**0.340**	55920
4	0.435	228526	0.285	64488
5	0.433	248692	0.312	71971
6	0.438	274378	0.308	84576
Baseline	0.352	483846	0.251	121625

GRAS' Parameters. One of the important steps in building statistical stemmers is find the best of set of values for the parameters [28]. For GRAS, although the authors provide some general hints in [22] (i.e., λ to be the average word length for the language concerned, $\alpha = 4$ and $\delta = 0.8$), but we decided to run a set of experiments to empirically identify the best vales for the these parameters.

From the computational complexity perspective, λ is the most important parameter, as it directly affects the complexity of the graph decomposition step. In our experiments, we varied the value of λ from 3 to 7 (Sorani's average word length is 5.6; for Kurmanji it is 4.8 [5]). Moreover, since running the algorithm with $\lambda = 3$ and $\lambda = 4$ on the full version of our Sorani lexicon (generated from all documents in Pewan's Sorani corpus) exhausted our computational resources, for these cases we used reduced lexicons (generated from 10% and 25% of the documents, accordingly).

The results of this study is shown in Table 3. We would like to note that in the interest of space and due to its inferior performance, the results for $\lambda = 7$ are not included in this table.

Based on these numbers, the following observations can be made: (i) our implementation of the GRAS statistical stemmer generally improves the retrieval performance, (ii) while for Kurmanji the best outcome is achieved for ($\lambda = 3, \alpha = 2, \delta = 0.7$), Sorani's peak is reached at ($\lambda = 4, \alpha = 6, \delta = 0.9$).

5.2 Analysis

The MAP measure is useful to compare the *overall* performance of different IR systems. In order to better understand the behavior of these systems, a detailed analysis of the results is required. To this end, in the following we present a drill-down comparison of Jedar's and GRAS' outputs at their best-performing configuration.

Detailed Comparison. Figure 4 depicts the precision curves at the standard 11 recall points. It clearly demonstrates the facts that (i) both Jedar and GRAS improve the IR performance at all recall levels, (ii) the gains from Jedar and GRAS are comparable.

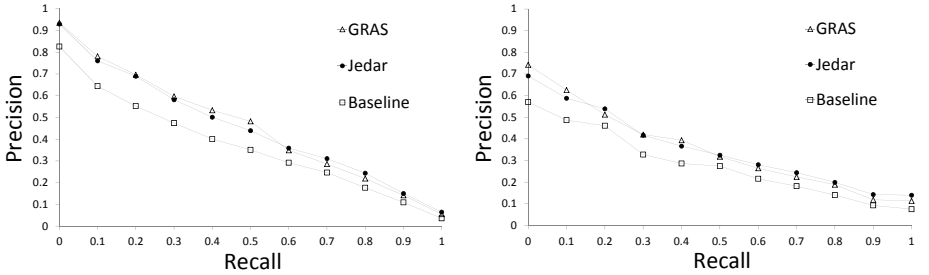
Table 3. Tuning GRAS' Parameters

Parameters			Sorani		Kurmanji	
λ	α	δ	MAP	Lexicon Size	MAP	Lexicon Size
3	2	0.7	0.356	393631	**0.341**	43704
		0.8	0.387	395257	0.280	15290
		0.9	0.400	404448	0.280	15290
	4	0.7	0.422	404137	0.280	15290
		0.8	0.387	395257	0.280	15290
		0.9	0.400	404448	0.280	15290
	6	0.7	0.356	393631	0.280	15290
		0.8	0.387	395257	0.280	15290
		0.9	0.407	412398	0.280	15290
4	2	0.7	0.445	343658	0.305	63316
		0.8	0.364	315986	0.313	63563
		0.9	0.364	315986	0.304	66333
	4	0.7	0.364	315986	0.300	70634
		0.8	0.364	315986	0.300	70894
		0.9	0.364	315986	0.319	72642
	6	0.7	0.427	361278	0.300	73678
		0.8	0.432	352719	0.305	73943
		0.9	**0.448**	352523	0.314	75597
5	2	0.7	0.440	198860	0.296	79240
		0.8	0.415	161743	0.296	79324
		0.9	0.415	161743	0.301	80177
	4	0.7	0.415	161743	0.294	83627
		0.8	0.415	161743	0.296	83714
		0.9	0.415	161743	0.308	84436
	6	0.7	0.415	161743	0.294	85466
		0.8	0.415	161743	0.298	85529
		0.9	0.415	161743	0.314	86343
6	2	0.7	0.409	232014	0.292	93144
		0.8	0.409	231975	0.293	93197
		0.9	0.409	231975	0.294	93543
	4	0.7	0.409	231975	0.305	95794
		0.8	0.409	231975	0.308	95805
		0.9	0.409	231975	0.308	96224
	6	0.7	0.409	231975	0.306	97002
		0.8	0.409	231975	0.306	97047
		0.9	0.409	231975	0.304	97488
Baseline			0.352	483846	0.251	121625

Given GRAS' reasonable computational cost and its language-independent nature, this can mean that GRAS is the favorable option.

Query-Level Analysis. We also carried out a query-level examination of the results and identified three distinct groups among the queries. Below, we enumerate these groups and present an example for each one:

- *GRAS outperforming Jedar*: for example for Q_{21} in the Sorani experiments, while GRAS correctly puts the words سووریە *Soorya* and سووریا *Sooryaa* (different variations of the country name "Syria") in one cluster, Jedar over-stems the word سووریە *Soorya* to سوور *Soor* (the color name "red") which is obviously irrelevant and entails ambiguity.
- *Jedar outperforming GRAS*: for instance in the Sorani version of Q_{22} , GRAS puts the named entity تووڕەج *Tooraj* (first name of a local photographer) into an irrelevant cluster with the stem تووڕەبوون *Toorraboon* "resent".



(a) On the Sorani Collection (b) On the Kurmanji Collection

Fig. 4. PR-Graphs for the Best-Performing Configurations of Jedar and GRAS

– *Stemming unhelpful*: e.g., for query Q_{10} in the Kurmanji experiments, both stemming approaches result in performance degradation, compared to the baseline approach in which no-stemming is applied. This is because both Jedar and GRASS consider the composite named entity `Hikûmeta Herêma Kurdistanê` (*Kurdistan Regional Government*) to be three independent words and stem them separately, leading to retrieval of irrelevant documents.

6 Conclusions and Future Work

In this paper we presented Jedar, the first rule-based stemmer for Sorani Kurdish and Kurmanji Kurdish. We also introduced our implementation of GRAS [22], a recent proposal for statistical stemming. After fine-tuning their parameters, these stemmers were used to empirically study the effectiveness stemming for Kurdish IR.

Our results show that: (i) both Jedar and GRAS can significantly improve the performance of Kurdish IR systems, (ii) the rule-based approach and the the statistical stemmer approach perform comparably well, (iii) overall, the shorter stem lengths (i.e., 3,4) seem to be more effective.

In future, we plan to propose solutions to fix some of the systematic stemming errors that we highlighted in the analysis section (e.g., over-stemming and mishandling of named entities). Comparing the performance of these stemmers against N-grams is another avenue for future work.

References

1. Bacchin, M., Ferro, N., Melucci, M.: A Probabilistic Model for Stemmer Generation. Information Processing and Management 41(1), 121–137 (2005)
2. Blau, J.: Méthode de Kurde: Sorani. Harmattan (2000)
3. Braschler, M., Ripplinger, B.: How Effective is Stemming and Decompounding for German Text Retrieval? Information Retrieval 7(3-4), 291–316 (2004)
4. Esmaili, K.S., et al.: Building a Test Collection for Sorani Kurdish. In: Proceedings of IEEE AICCSA (2013)
5. Esmaili, K.S., Salavati, S.: Sorani Kurdish versus Kurmanji Kurdish: An Empirical Comparison. In: Proceedings of the 51st Annual Meeting of ACL (2013)

6. Esmaili, K.S., Salavati, S., Datta, A.: Towards Kurdish Information Retrieval. ACM TALIP (to appear, 2013)
7. Gautier, G.: Building a Kurdish Language Corpus: An Overview of the Technical Problems. In: Proceedings of ICEMCO (1998)
8. Haig, G., Matras, Y.: Kurdish Linguistics: A Brief Overview. Language Typology and Universals 55(1) (2002)
9. Harman, D.: How Effective is Suffixing? JASIS 42(1), 7–15 (1991)
10. Hassanpour, A., et al.: Introduction. Kurdish: Linguicide, Resistance and Hope. International Journal of the Sociology of Language 217, 1–8 (2012)
11. Hull, D.A.: Stemming Algorithms: A Case Study for Detailed Evaluation. Journal of the American Society for Information Science 47(1), 70–84 (1996)
12. KLPP. Kurdish Language Stemmers, http://klpp.github.io/
13. KLPP. The Pewan Test Collection, http://klpp.github.io/
14. Krovetz, R.: Viewing Morphology as an Inference Process. In: Proceedings of ACM SIGIR 1993, pp. 191–202 (1993)
15. Lovins, J.B.: Development of a Stemming Algorithm. MIT Information Processing Group, Electronic Systems Laboratory (1968)
16. MacKenzie, D.N.: Kurdish Dialect Studies. Oxford University Press (1961)
17. Majumder, P., Mitra, M., Pal, D.: Bulgarian, hungarian and czech stemming using YASS. In: Peters, C., Jijkoun, V., Mandl, T., Müller, H., Oard, D.W., Peñas, A., Petras, V., Santos, D. (eds.) CLEF 2007. LNCS, vol. 5152, pp. 49–56. Springer, Heidelberg (2008)
18. Majumder, P., Mitra, M., Parui, S.K., Kole, G., Mitra, P., Datta, K.: YASS: Yet Another Suffix Stripper. ACM TOIS 25(4), 18 (2007)
19. MG4J. Managing Gigabytes for Java, http://mg4j.dsi.unimi.it/
20. Monz, C., De Rijke, M.: Shallow Morphological Analysis in Monolingual Information Retrieval for Dutch, German, and Italian. In: Evaluation of Cross-Language Information Retrieval Systems, pp. 262–277 (2002)
21. Paice, C.D.: An Evaluation Method for Stemming Algorithms. In: Proceedings of ACM SIGIR 1994, pp. 42–50 (1994)
22. Paik, J.H., Mitra, M., Parui, S.K., Järvelin, K.: GRAS: An Effective and Efficient Stemming Algorithm for Information Retrieval. ACM TOIS 29(4), 19 (2011)
23. Porter, M.F.: An algorithm for suffix stripping, pp. 313–316. Morgan Kaufmann Publishers Inc. (1997)
24. Porter, M.: Snowball: A Language for Stemming Algorithms (2001)
25. Samvelian, P.: When Morphology Does Better Than Syntax: The Ezafe Construction in Persian. Ms., Université de Paris (2006)
26. Samvelian, P.: A Lexical Account of Sorani Kurdish Prepositions. In: Proceedings of International Conference on Head-Driven Phrase Structure Grammar, pp. 235–249 (2007)
27. Samvelian, P.: What Sorani Kurdish Absolute Prepositions Tell Us about Cliticization. Texas Linguistic Society IX, p. 265 (2007)
28. Smirnov, I.: Overview of Stemming Algorithms. Mechanical Translation (2008)
29. Walther, G.: Fitting into Morphological Structure: Accounting for Sorani Kurdish Endoclitics. In: The Proceedings of the Eighth Mediterranean Morphology Meeting (2011)
30. Walther, G., et al.: Fast Development of Basic NLP Tools: Towards a Lexicon and a POS Tagger for Kurmanji Kurdish. In: Proceedings of the 29th International Conference on Lexis and Grammar (2010)
31. Walther, G., Sagot, B.: Developing a Large-scale Lexicon for a Less-Resourced Language. In: SaLTMiL's Workshop on Less-resourced Languages (LREC) (2010)
32. Xu, J., Croft, B.: Corpus-based Stemming Using Cooccurrence of Word Variants. ACM TOIS 16(1), 61–81 (1998)
33. Xu, J., Fraser, A., Weischedel, R.: Empirical Studies in Strategies for Arabic Retrieval. In: Proceedings ACM SIGIR 2002, pp. 269–274 (2002)