DIAC$^+$: A Professional Diacritics Recovering System

Dan Tufiş, Alexandru Ceauşu
Institute for Artificial Intelligence, Romanian Academy
"13 Septembrie", 13,050711, Bucharest 5
tufis@racai.ro, aceausu@racai.ro

Abstract

In languages that use diacritical characters, if these special signs are stripped-off from a word, the resulted string of characters may not exist in the language, and therefore its normative form is, in general, easy to recover. However, this is not always the case, as presence or absence of a diacritical sign attached to a base letter of a word which exists in both variants, may change its grammatical properties or even the meaning, making the recovery of the missing diacritics a difficult task, not only for a program but sometimes even for a human reader. We describe and evaluate an accurate knowledge-based system for automatic recovering the missing diacritics in MS-Office documents written in Romanian. For the rare cases when the system is not able to reliably make a decision, it either provides the user a list of words with their recovery suggestions, or probabilistically choose one of the possible changes, but leaves a trace (a highlighted comment) on each word the modification of which was uncertain.

1. Introduction

Spell checking is one of the oldest natural language processing applications that is used on large scale. Most of the spell checkers, rely on large word-form lexica and are designed to detect spelling errors and suggest possible corrections. In keyboarding a text most users make typing errors, the majority of them due to transposing characters, pressing the key next to the intended one, omitting a character, inserting an unnecessary character, or omitting a space between words. A different class of errors (called cognitive errors) refers to those situations where the user does not know the correct orthography of a word and uses a plausible near-miss. Yet, there is another important aspect of spelling verification and correction which, although does not occur in English, is relevant for almost all European languages (see (Mihalcea, 2002), Table 1): restoring the diacritics wherever they are missing.

Having an automatic procedure for restoring the missing diacritics is worthy not only for old valuable texts stored in electronic form, but also for contemporary electronic texts as they continue to be produced in non-diacritical form. The reasons for this could be many, including the lack of localized and standardized keyboards. Ergonomic factors can also be mentioned (if someone is supposed to press more than two keys to get a diacritical character, then, mainly in informal communication (e.g. e-mail), he/she will probably take the easiest one-stroke solution).

The problem of diacritics restoration has been, and continues to be, addressed by various researchers with respect to different languages. There are two major approaches in solving this problem, one based on words and the other one based on characters. While the word-based approaches are, in general, informed systems, relying on lexica and language models (thus being language dependent), the character-based methods are language independent uninformed algorithms that do the job based on statistical information on n-grams extracted from given training data. Each approach has merits and drawbacks; the informed word-based systems require large lexical resources (which are never completely covering any new text) and their maintenance, additional processing tasks (e.g. tokenization, tagging) and, thus, need more time to finish the job. The character based systems are easy to implement, need only raw training data for the languages of interest and are very fast. On the other hand, for languages where the diacritics have grammatical and/or semantic role, the word-based systems are much more reliable than the character-based systems. For such languages, the lack of, or mistakenly added diacritical signs may be extremely annoying especially in texts meant for publication. For languages where the absence of required diacritics is context-independent detectable, a character-based (n-gram) approach can do the work almost equally accurate but much faster and with little development effort.

The decision on the approach to follow in the development of a diacritics restoration program depends on many factors: the grammatical and/or semantic role of the diacritics in the language of interest, the availability of adequate language resources, the required processing speed, the users' requirements and needs.

We describe and evaluate an accurate knowledge-based system, called DIAC$^+$, for automatic recovering the missing diacritics in MS-Office documents written in Romanian. The out of lexicon words (usually, very rare) are processed by a character-based back-off procedure. The present system builds on our previous work (Tufiş and Chiţu, 1999), extending the former DIAC system in many ways and significantly improving its performance and appearance.

2. Related work

Word-based implementations of diacritics restoration programs have been described, among the others, in (El-Bèze et al, 1994), (Yarowsky, 1994), (Spriet & El-Bèze (1997), (Simard, 1998), (Tufiş & Chiţu, 1999), etc.

El-Bèze and his colleagues (1994) use POS-tagging of French texts to exploit the contextual information and N-gram statistics to decide whether or not an accent has to be added for the word under investigation. In a follow-up paper, Spriet and El-Bèze (1997) describe the use of an N-gram model on parts-of-speech for re-accentuation of French texts. They evaluate their method on a 19,000 word test corpus consisting of news articles and obtain a 99.31% accuracy. In this corpus, only 2.6% of the words were unknown, among which 89.5% did not need accents. The resulting error rate (0.3%) accounts for nearly one half of the total error rate, but is so small that it is not worth trying to guess accentuation for unknown words (cf. Zweigenbaum & Grabar, 2002).

Yarowsky (1994) addresses this problem for Spanish (mainly) and French but instead of POS tagging, he reports the best performance with a decision-list framework.

Simard (1998) also uses the POS tagging technology, but as all the previously mentioned systems, leaves the out-of-lexicon words untouched.

The methodology described in (Tufiş & Chiţu, 1999) is similar to the one presented in (Simard, 1998), but the system, developed for Romanian language, have a better success rate, although, compared to French, Romanian makes more intensive use of diacritical signs and their absence creates much more difficulties. As before, this system ignores the unknown words.

The **character-based** approaches became quite popular lately, mainly because of their simplicity, language independence, good performance and easy to get training data (which is simply, raw texts containing the required diacritics). The expensive wide-coverage lexicons are not required by these methods.

Mihalcea (2002) addresses the restoration of diacritical characters in a Romanian electronic dictionary using an n-gram model and made experiments with a memory-based learning system (TIMBL) and a decision tree classifier (C4.5). She reports in each case very good precisions (letter-based), with average accuracy beyond 99%.

Using the same evaluation data as used by Mihalcea (2002) Bobiceva (2008) describes another letter-based implementation of the diacritics recovery for Romanian. She applied a statistical method used in text data compression (PPM- prediction by partial matching) and shows that her method obtains comparable results to Mihalcea's. Both methods have hard times with dealing with the *a-ă* (the average precision is 96,15%) mainly when the respective letters are word final[1].

Zweigenbaum and Grabar (2002) describe a system specifically designed for recovering the various diacritical versions of 'e' (*é, è, ê, ë*) in specialized French lexicon (the French version of MeSH). They use two different methods: a finite state transducer and Brill's tagger to learn contextual transformation rules for the letter 'e' (each proper word, containing an 'e' is split into its constituent letters which are considered the tokens to be tagged).

When dealing with resource-scarce languages, as African languages described in (Wagacha et. al. 2006) or (De Pauw et. al. 2007), the character-based methods hardly have a choice competitor.

3. Diacritics in Romanian

Romanian language has 5 diacritical characters: *ă,â,î,ş* and *ţ* (plus their uppercase variants). A text missing the diacritics will usually have these characters substituted by *a* (for both *ă* and *â*), *i*, *s* and *t* respectively. For a significant part of the words with the diacritics stripped-off their recovering is deterministic, because the non-diacritical variants of those words are not legal lexemes of Romanian. But in most of the cases, the absence of diacritics creates genuine ambiguity, hard to resolve sometimes even for a human (when given only a limited context).

Here are some examples of strings with missing diacritics that are not valid Romanian words (the real word and its translation are specified between parentheses):

A) padure (*pădure* - forest), tufis (*tufiş* - bush), cantar (*cântar* - balance), carare (*cărare* - pathway), casmir (*caşmir* - cashmere), macar (*măcar* - at least), fara (*fără* - without), cati (câţi - how many), etc.

Such strings, which could be unambiguously recovered by relying on a adequate lexicon, are called *U-words.*

To exemplify the ambiguity caused by the lack of diacritics, let us consider the string *fata*. In a text where the diacritics were removed, this string could stand for any of the following words:

B) *fata* – the girl, *fată* – a girl; or (about animals) gives birth , *fâţa* – the quick-swimming little fish/the coquette, *fâţă* – a quick-swimming little fish/a coquette, *faţa* – the face, *faţă* – a face, *făta* – (about animals) to give birth; gave birth, *fătă* – (about animals) just gave birth.

All the strings of the *fata* type above (i.e which could stand for more than one diacritical or non-diacritical word) are referred to in the following as ambiguous stripped words, or *A-words.* The strings that are neither U-words nor A-words are simply referred to as words.

We found that the morpho-syntactical information disambiguates most A-words. Yet, there exist subsets of A-words for which morpho-syntactic descriptions are identical and diacritics restoration distinction could be made only based on meaning:

C) *fata* (Ncfsry) – the girl, *fâţa* (Ncfsry) – the quick-swimming little fish/the coquette, *faţa* (Ncfsry) – the face.

These words, which we call *S-words*, require sense disambiguation. The S-words are a subset of A-words.

The Table 1 displays data we extracted from our reference corpora. The journalism corpus consists of articles from the weekly magazine “Agenda” from Timişoara (years 2003-2006). The juridical corpus is a collection of around 6000 Romanian documents extracted from the Jrc-Acquis corpus (Steinberger et al., 2006). The

[1] In Romanian this is the most difficult case, since feminine nouns and adjectives ending in "a" are definite forms while when ending in "ă" they are indefinite forms. Also, the verb-final "a/ă" distinguishes among tenses (present vs. simple perfect). When embedded, this alternation frequently changes the meaning of the word (par=pole, versus păr=hair or pear tree).

part-of-speech annotation was made with our tiered tagger in order to reduce as much as possible the number of tagging errors. The total number of words shown in Table 1 (line 1) does not include numbers or tokens containing one or more digits, proper names, foreign words (tagged by the X tag), abbreviations (tagged by the Y tag), dates (tagged by the DATE tag) and punctuation. From the total number of tokens in the mentioned texts, the discarded tokens account for 36% and 26% respectively. The big difference between the number of discarded items in the two corpora is due to the fact that journalism corpus contains lots of numbers (in the "Sport" sections one can find scores, minutes, timings, distances etc) dates, foreign words and abbreviations. These categories are not significant for the diacritics restoration problem because, in the vast majority of cases, they do not contain diacritical signs. However, the proper names in Romanian are words that might contain diacritics, thus being relevant for the diacritics restoration task. Yet, in the juridical corpus, although the names are quite frequent, none of them contained diacritics. Thus, in order to make a meaningful comparison among the two register data, we excluded the proper names from our analysis.

There are two different figures for S-words, depending on what tagset was used in POS tagging: a reduced tagset (Ctag-set in line 5) and the lexicon morpho-syntactic descriptors tagset (MSDtag-set in line 6). The two figures demonstrate that the diacritics restoration is more accurately done when the system has access to more linguistic contextual information. On the other hand, in general, using a reduced tagset as compared to a large one, increases the tagging accuracy, which is vital for our approach in diacritics restoration. The Ctag-set and MSDtag-set and the way we solved the tension between tagset cardinality and tagging accuracy are briefly discussed in section 4.2.

Corpus	**Journalism**	**Juridical**
1. Words	6680448	3511093
1* Characters	37008236	21404666
2. Words with diacritics (of 1.)	2004763 (30,01%)	1026385 (29,23%)
2*. Diacritics	2351220	1192875
3. U-words (of 2.)	238132 (11,88%)	175822 (17,13%)
4. A-words (of 2.)	1766631 (88,12%)	850563 (82,87%)
5. S-words (Ctag-set, of 4)	58420 (3,31%)	38323 (4,51%)
6. S-words (MSDtag-set, of 4)	24916 (1,41%)	16463 (1,94%)

Table 1. The distribution of the words with diacritics in texts of different registers

As one can notice in the table above, in regular Romanian texts, almost one third of the words contain at least one diacritical character (30% of the words in the journalism data contain on average 1,17 diacritical signs, while 29% of the words in the juridical texts contain on average 1,16 diacritical signs). Out of the diacritical words only a small percentage are U-words (12% in the journalism data and 17% in the juridical texts). That is, in an ideal setting, with a fully coverage dictionary available and a text with no typographical error other than the missing diacritics, about 25% (#A-words/#Words) of the total number of words in a running Romanian text would remain ambiguous. In a more realistic setting, this figure is significantly higher because no dictionary fully covers any possible text and most texts contain typing errors (other than missing diacritics). In our supposedly error free data we identified 72722 (1.09%) typing errors in the journalism texts and 29387 (0.84%) typing errors in the juridical texts. Below we list the main categories of errors:

a) even if a word contains diacritics, it might not contain all of the necessary ones (e.g. "invăţămant"[2] vs. "învăţământ", "lacătuş" vs. "lăcătuş" etc.);

b) even if a word contains diacritics, one or more of them might be wrong (e.g. "sărmă" vs. "sârmă", "câtre" vs. "către", "neâncăpător" vs. "neîncăpător" etc.)

c) even if a word contains diacritics, they might not be in accordance with the current orthography of Romanian (e.g. "considerînd" vs. "considerând", "curînd" vs. "curând" etc.)

d) the words (with or without diacritics) might be misspelled (e.g. "înopta" vs. "înnopta", "indenmizaţie" vs. "indemnizaţie", "compensdiu" vs. "compendiu" etc.) or miss-tokenized (e.g. "5%pentru" vs. "5% pentru" etc.)

e) a lexical token might be distorted by a combination of the above cases, making it very difficult to be recovered.

One could argue that a traditional spell-checker could identify these error-cases and a user might fix them, but at least for Romanian, this is not entirely so, because of what we called A-words (words which remain legal words of the language even after the diacritics removal, e.g. "peste" (*over*) versus "peşte" (*fish*), "scoală" (*wake up*) versus "şcoală" (*school*), "barca" (*the boat*) versus "barcă" (*a boat*) etc. A standard spell-checker (such as A-spell or the one included into MS Office) would not even detect the A-words as possibly problematic.

In a traditional spell-checker solution, the "out of the dictionary" words are highlighted and the user is expected to select one correction from a list of possible choices (which might not include the proper correction). In our approach, most of the corrections (all but the S-words) are automatically performed without user going through each individual token. While the automatic procedure is practically done in no time, the manual procedure is error prone and even when assisted by a spell-checker might require hours, days or even months for very large textual data. For the S-words occurring in a text, the system behaves like a spell-checker, i.e. it requires the user to make a choice, out of a list of contextually plausible corrections. A contextual plausible correction should

[2] All the examples in this paper are extracted from the corpora described in Table 1.

comply with the linguistic restrictions specified by the morpho-syntactic description associated to the respective S-word. For instance, if the S-word "fata" was tagged as a feminine noun, in a direct case, definite singular form, the plausible solutions are "fata" (the girl), "fața" (the face), "fâța" (the quick-swimming little fish/the coquette) all characterized by the same morpho-lexical attributes as the original S-word. All the other variants (fată, față, fâță, făta, fătă) should be ignored due to different morpho-syntactic descriptors (indefinite nouns or verbs).

The last two lines in Table 1 show that when we use a tag-set of finer granularity (614 tags in the MSDtagset versus 92 tags in the Ctagset) the number of S-words (the tokens for which the diacritical forms cannot be deterministically recovered) is more than twice less than before. We noticed that the majority of the S-words in this case are genuine spelling errors, very few of them requiring sense disambiguation.

In the next sections we will briefly describe the underlying technologies used by our diacritics restoration system DIAC$^+$, provide an evaluation and few details on its implementation and conclude with a comparison between DIAC$^+$ and its ancestor described in (Tufis, Chitu 1999).

4. Text pre-processing

Since the DIAC$^+$ was designed to work with MS formatted documents, the system extracts the textual data from the input file and stores it in an internal format adequate for our pre-processing tools, using as database a full-text search engine – Lucene[3]. The textual data extracted from the input file is tokenized and tiered-tagged, thus creating a linguistic knowledge space for the current text within which the proper restoration of diacritics takes place.

4.1 Tokenization

The tokenizer is a program that identifies within the input text the elementary processing units called lexical tokens. A lexical token usually corresponds to the generally accepted idea of a word, namely a sequence of characters delimited by white spaces. However, several words may form a natural single unit (such as "*pentru că*" – because) or on the contrary, a sequence of characters delimited by white spaces may be split into distinct lexical units (such as "*dă-mi-le*" – you_(singular) give to_me them = give them to me). The tokenizer also recognizes dates expressed in a large variety of formats (1 ianuarie, 1999; 01/01/99; 01-ian-99, etc), abbreviations (*dl, dna, dra*, *dr.* etc.), various types of punctuation, etc.

4.2 Tiered tagging

In highly inflectional languages, encoding the morpho-lexical properties of the word-forms requires a large set of description codes. The Multext European project in co-operation with EAGLES Lexical Specification Group developed a set of recommendations for the languages in Western Europe. Starting with these specifications, the Multext-East Copernicus project further developed them so that to account for the specificity of six other languages from Central and Eastern Europe – Bulgarian, Czech, Estonian, Hungarian, Romanian and Slovene (see http://nl.ijs.si/ME/). The set of morpho-syntactic descriptors (MSDs) specific to Romanian contains 615 codes.

It is well known that the larger the tag-set, the larger the training corpora needed and unfortunately this is not a linear dependency. To avoid severe data sparseness and accuracy degradation, a huge amount of manual work would be necessary for building appropriately large training corpora.

Tiered tagging (Tufiş, 1999, 2000) is a two-stage technique addressing the issue of data-sparseness. In general terms, tiered tagging uses a hidden tagset (we call it Ctag-set) of a smaller size (in our case 92 tags) on the basis of which a language model (LM) is built. This LM serves for a first level of tagging. Then, a second phase replaces the tags from the small tagset with contextually the most probable tags from the large tagset (we call it MSDtag-set) which contains 615 tags (MSDs). The fundamental idea in using the tiered tagging approach is that the attribute values in a MSD and the word-form are not independent. That is to say, having a MSD-based word-form lexicon, from a word-form and a subset of attribute-value pairs one could, in the vast majority of cases, deduce all the rest of the feature-values pairs characterizing the current word-form. In (Tufis, 1999) we call this property the MSD-*recoverability.* The subset of the features in the MSDtag-set having the recoverability property represents the set of attributes in terms of which the Ctag-set is defined. In (Tufis, 1999) we provided an algorithm to construct the Ctag-set from a MSD-based lexicon. In (Tufis, 2000) we demonstrated that the algorithm is language independent and that the tiered-tagging approach is working very well for a completely different language than Romanian. In (Tufis, Dragomirescu, 2004) we presented a further enhanced version of the Ctag-set automatic design and demonstrated its effectiveness on six languages (Czech, English, Estonian, Hungarian, Romanian and Slovene).

The lexicon, underlying the induction of the Ctag-set and backing-up the tiered tagging approach, contains the words annotated with the MSD tags, an entry having the form: <*word*> <*lemma*> <*msd*>. For Romanian, this lexicon, referred to in the following as LEX, contains more than 800,000 entries.

For a small number of the C-tags, the recovering process can face some ambiguities which have to be solved by using additional knowledge resource. In (Tufis, 1999) this new resource is a set of hand-written contextual disambiguation rules. The applicability of both the deterministic and the rule-based recovering is limited only to the words recorded in the MSD tag-set lexicon. We replaced the second phase of the tiered tagging process with a maximum entropy-based MSD recovery (Ceauşu, 2006). In this approach, the rules for Ctag to MSD

[3] http://www.apache.org/dyn/closer.cgi/lucene/java/

conversion are automatically learnt from the corpus and their application does not require looking-up the MSD tag-set lexicon. Therefore, even the Ctags assigned to unknown words can be converted into MSD tags. If an MSD-lexicon is available, replacing the Ctags for the known words by the appropriate MSD tags is almost 100% accurate.

5. Diacritics insertion

The overall architecture of DIAC$^+$ is shown in Figure 1.

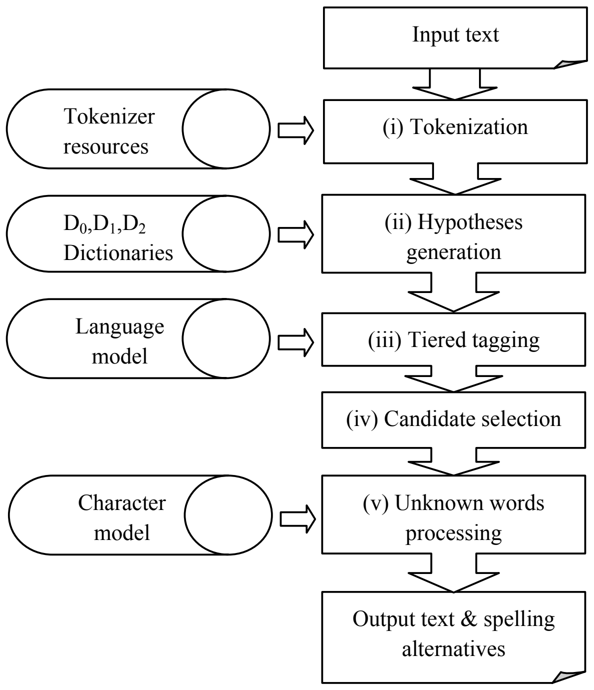

Input text
Tokenizer resources
(i) Tokenization
D_0,D_1,D_2 Dictionaries
(ii) Hypotheses generation
Language model
(iii) Tiered tagging
(iv) Candidate selection
Character model
(v) Unknown words processing
Output text & spelling alternatives

Figure 1. General architecture of DIAC$^+$

From the LEX lexicon, mentioned in the previous section, the system derives a diacritical words only lexicon (D_0), and a diacritics stripped-off lexicon (D_1) which are used to generate the hypotheses search space for the current text. Additionally, the system builds on the fly a list of words in the current text which are not in the previous dictionaries but which could be considered typographical errors (D_2):

- D_0 dictionary is the subset of LEX containing all the words containing at least one diacritical character;

- D_1 dictionary is the diacritics stripped-off version of LEX; one should bear in mind that the entries from D_0 corresponding to A-words will differ among each other only by POS information

- D_2 dictionary contains words in the current text which are neither in D_0 nor in D_1 and which are suspected of being typing errors; they are derived from the words in $D_0 \cup D_1$ differing by plus or minus one character or by switching two consecutive characters (additionally, the switched characters should be neighbors on the keyboard).

The procedure for automatic insertion of diacritics in Romanian texts has four stages:

(i) TOKENIZATION. The input text is segmented into lexical tokens according to the rules specified as external resources.

(ii) HYPOTHESES GENERATION. In the hypotheses generation step, a word is first searched in the union of D_0 and D_1 dictionary because in a text without diacritics or with partial diacritics one cannot be sure if a word is in its regular form or not unless contextual information is available.

If the word cannot be found in the union of D_0 with D_1 it is searched in the D_2 dictionary. A word which is not found in any of the system's lexicons is considered unknown and irrecoverable by the word-based approach, and its processing is left in charge of a character-based recovery module.

In this step, a word *W*, occurring in the current text, may be associated with several entries in the LEX word-form lexicon and as such it will be associated with a set of pairs <$surface\text{-}form_k$ MSD_k> provided that the diacritics stripped-off versions of the $surface\text{-}form_k$ and of *W* are identical. The information provided by the next tagging step will be used to filter this set and eventually to select the single contextually correct <$surface\text{-}form_i$>.

(iii) TIERED TAGGING. The text is tiered-tagged (tagged with the reduced tag-set, then each C-tag is mapped to a MSD-tag by the ME-tagger (Ceauşu, 2006); for this stage, only the MSDs from the hypothesis generation step are taken into consideration). In the case of unknown words the tagger chooses the best alternative resulted from the maximum entropy model. For tagging texts with partial or missing diacritics we used a special HMM language model in which the transition probabilities were computed from the regular training corpora (i.e. with diacritics) and the emission probabilities were computed from the diacritics stripped-off training corpora. This way the ambiguity classes for the words in the probabilistic lexicon and their respective POS lexical probabilities were modified, but the transition probabilities remained unchanged. For instance, the two unambiguous words *peste/*Spsa (eng. over) and *peşte/*Ncms-n (eng. fish) in the diacritics stripped-off training corpora will be represented by the same token type (*peste*) which in this case will become POS ambiguous (Spsa or Ncms-n). It is obvious that the spurious ambiguities created by the lack of diacritics degrade the tagging accuracy, but as discussed in (Tufis, Chitu, 1999) not all tagging errors are harmful for the diacritics restoration process.

(iv) CANDIDATE SELECTION. The U-words are replaced with their diacritical counterpart. The A-words which are not S-words are replaced by the *surface-form* identified by the MSD assigned by the tagger to the respective A-word. For the S-words, depending on the DIAC$^+$ variant (see further) either the user is presented with a list of contextually meaningful choices or the replacement is automatically done based on lexical probabilities or some probabilistic preferences.

(v) UNKNOWN WORD PROCESSING is used as backup for the candidate selection stage where no equivalent word-form was found in the lexicon. This case is quite rare –

very few words are not covered by the 800000 entries lexicon. The stage of unknown word processing can be designed to work in parallel with the stage of candidate selection. For processing unknown words, we used a character-based N-gram model similar to the one used in (Mihalcea, 2002).

Model order	Perplexity	Accuracy (no spaces)	Model size
2-gram	12.42	93.67%	20.8 KB
3-gram	9.72	95.52%	223 KB
4-gram	7.11	97.72%	1.29 MB
5-gram	5.77	98.59%	4.82 MB
6-gram	5.29	98.79%	13.1 MB
7-gram	5.17	98.84%	27.7 MB
8-gram	5.18	98.85%	48.4 MB

Table 2. Evaluation of several character models for unknown words processing

We opted to use Viterbi estimation with a 5-gram character model to find the most probable string for the unknown word. We used SRILM - SRI Language Modeling Toolkit (Stolcke, 2002) to train several character models. The training corpus contained 5124277 characters (including spaces) in 48308 sentences and the test corpus has 613234 characters in 6411 sentences. Table 2 displays a comparison of the perplexity, accuracy and size for the models of different order.

6. Evaluation

For the evaluation purposes we used a reference corpus R, containing about 118,000 words and about 502,000 characters. The reference corpus was hand tagged and lemmatized. We removed all the diacritics, from R but preserved the original tagging. This version of R is what we call the idealized DIAC$^+$ tagged text (TT): it has no tokenization or tagging errors, and no diacritical character is present in the text. Running DIAC$^+$ on TT provided us with an evaluation of the upper-limit of the system's accuracy (when perfect tagging is available).

For a more realistic setting we further removed from TT the associated tags getting a raw tokenized text (RT) on which we applied the processing chain (tagging with the reduced tag-set, mapping the C-tags to MSD and DIAC$^+$). In both of these experiments DIAC$^+$ was used without any user interaction (that is with the S-words automatically dealt with).

The results of these evaluations are synthesized in Table 3. Unlike the statistics in Table 1, here, no tokens were removed from the evaluation.

In order to asses the diacritics insertion accuracy we developed a baseline system. The baseline system was built using the Agenda corpus (10 million tokens). The system uses a dictionary of non-diacritical forms with their valid counterparts ordered by the number of occurrences. The baseline system replaces the non-diacritical form with the most frequent word-form. The *correct surface forms* differences in the two experiments (1,27%) can be ascribed entirely to the tagging errors, but as mentioned before not all the tagging errors generate diacritics restoration errors. A significant part of the incorrect surface forms were S-words (321), which should have received the user attention and choice.

In Table 4 we show the evaluation results of the same experiments, but this time in terms of characters. As it can be seen, the accuracy evaluation on characters shows a performance of over 99% even for the baseline system. One should note that the error score in the character-based evaluation (0,6%) looks much better then the error score in the word-based evaluation (2,25%). This supports our previous intuition (see the footnote 4) that one could easily estimate one evaluation score (word or character based) knowing the other score and if the average word length is smaller than the average distance between two diacritical characters.

However, based on the log file they could be easily corrected. The rest of the incorrect surface forms resulted from tagging errors. Some of these tagging errors in Romanian are very difficult to solve in a limited context.

	DIAC on tagged text (TT)	DIAC on raw text (RT)	Baseline system
Tokens	117 909		
Words with diacritics	34745 (29,47%)		
S-words	361		
Unknown words	2130 (1,8%)		
Correct word-forms	116 810 (99,06%)	115 262 (97,75%)	113 491 (96,25%)
Incorrect word-forms	1 092 (0,94%)	2 609 (2,25%)	4 418 (3,75%)

Table 3. Word-form accuracy evaluation DIAC$^+$

Most of them refer to the tense value attribute (present and imperfect tenses) of verbs in the first class conjugation, the infinitive form of which ends in "a". Their resolution would require a post-tagging processing with an inspection of the neighboring clauses and an analysis of sequence-of-tenses (hoping that the neighboring verbs are not in the same conjugation class).

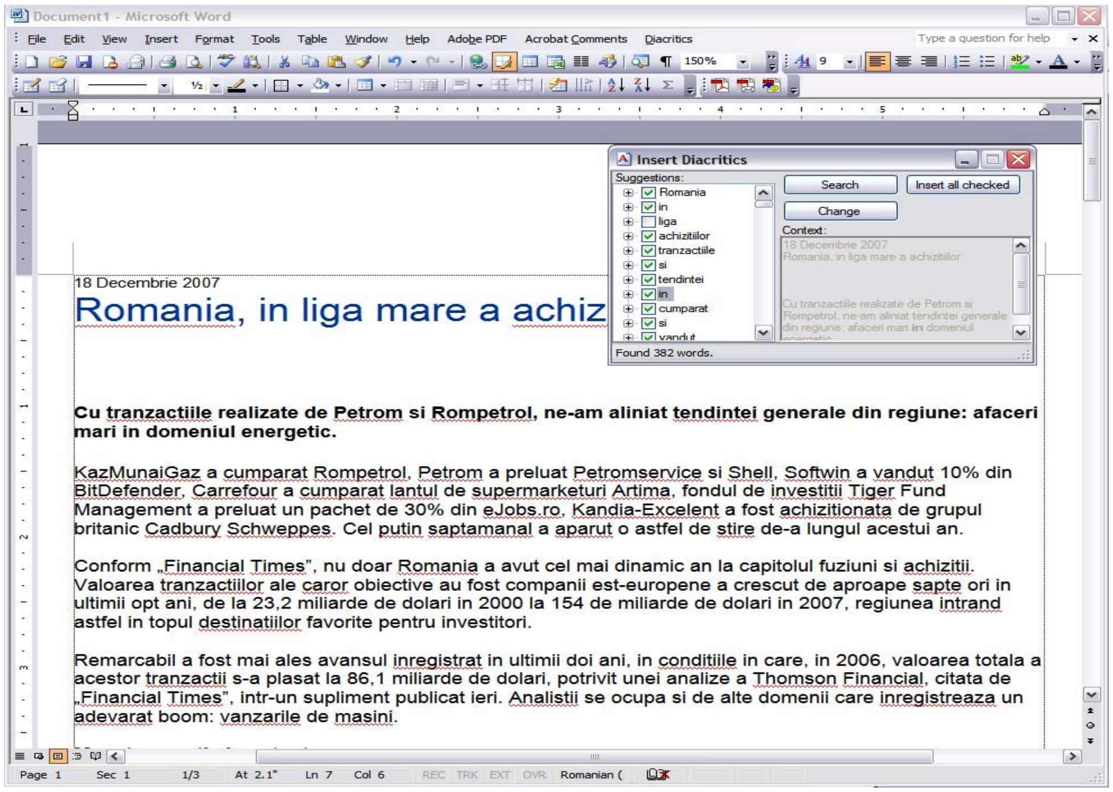
	DIAC on tagged text (TT)	DIAC on raw text (RT)	Baseline system
Characters (no spaces)	501735		
Diacritics	41144		
Correct characters (no spaces)	500400 (99.73%)	498764 (99.40%)	497096 (99,07%)
Incorrect characters (no spaces)	1335 (0,27%)	2971 (0,6%)	4639 (0,93%)

Table 4. Character accuracy evaluation DIAC$^+$

7. Implementation

The DIAC+ system is available in two implementations: as a web service, requiring a licensed access on our linguistic web-services platform for natural language processing and as a stand-alone variant, intended for local recovering of the diacritics in case of sensitive documents which the author might be reluctant to send via internet.

The web-service accepts a MS-Word document and autonomously decides on the required corrections . The S-words are corrected according to the user preferences, but a logfile is generated documenting each correction (initial word-form, possible replacements and the actual one). Optionally, the logfile can include for each replacement the sentence in which it was operated.


Document1 - Microsoft Word
File Edit View Insert Format Tools Table Window Help Adobe PDF Acrobat Comments Diacritics
Type a question for help
150%
Insert Diacritics
Suggestions:
Search
Insert all checked
Change
Context:
Page 1 Sec 1 1/3 At 2.1" Ln 7 Col 6 REC TRK EXT OVR Romanian (

☑ Romania
☑ in
☐ liga
☑ achizitiilor
☑ tranzactiile
☑ si
☑ tendintei
☑ in
☑ cumparat
☑ si
☑ vandut

18 Decembrie 2007
Romania, in liga mare a achizitiilor

Cu tranzactiile realizate de Petrom si Rompetrol, ne-am aliniat tendintei generale din regiune: afaceri mari in domeniul energetic

Found 382 words.

18 Decembrie 2007

Romania, in liga mare a achiz

Cu tranzactiile realizate de Petrom si Rompetrol, ne-am aliniat tendintei generale din regiune: afaceri mari in domeniul energetic.

KazMunaiGaz a cumparat Rompetrol, Petrom a preluat Petromservice si Shell, Softwin a vandut 10% din BitDefender, Carrefour a cumparat lantul de supermarketuri Artima, fondul de investitii Tiger Fund Management a preluat un pachet de 30% din eJobs.ro, Kandia-Excelent a fost achizitionata de grupul britanic Cadbury Schweppes. Cel putin saptamanal a aparut o astfel de stire de-a lungul acestui an.

Conform „Financial Times", nu doar Romania a avut cel mai dinamic an la capitolul fuziuni si achizitii. Valoarea tranzactiilor ale caror obiective au fost companii est-europene a crescut de aproape sapte ori in ultimii opt ani, de la 23,2 miliarde de dolari in 2000 la 154 de miliarde de dolari in 2007, regiunea intrand astfel in topul destinatiilor favorite pentru investitori.

Remarcabil a fost mai ales avansul inregistrat in ultimii doi ani, in conditiile in care, in 2006, valoarea totala a acestor tranzactii s-a plasat la 86,1 miliarde de dolari, potrivit unei analize a Thomson Financial, citata de „Financial Times", intr-un supliment publicat ieri. Analistii se ocupa si de alte domenii care inregistreaza un adevarat boom: vanzarile de masini.

Figure 2. Diacritics recovery in Microsoft Word 2003

The stand-alone version of the application is embedded into the Microsoft Office suite and complements the MS spell checker.

Once an MS-Office document is opened, pressing the *Search* button of the DIAC$^+$ interface will launch the entire processing chain (text extraction, tokenization, tiered-tagging and multi-criteria indexing) discussed in the previous sections. As a result, all the words in the current document, potentially requiring diacritics restoration, will be listed in the left pane (*Suggestions)* of the DIAC$^+$ interface as shown in Figure 2.

Each word-form listed in the *Suggestion* window is preceded by a '+' unfolding button and a "check" box. If the "check" box is checked-out (☑) the system signals that for the respective word it found a unique correction. Selecting a word-form in the "*Suggestions*" window, will bring-up in the "*Context*" window (left window in the DIAC$^+$ interface) the sentence containing the respective occurrence and scroll the document window highlighting the selected.

Pressing the "*Insert all checked*" will operate the respective corrections and in the "*Suggestions*" pane will remain only the words for which the system could not make an informed decision. These words are preceded by an unchecked box and pressing the '+' unfolding button will show the contextually possible diacritical forms. Each possible solution has a "check" button allowing the user to specify his option.

The system can correct a few typographical errors such as transposed characters, wrong typed characters, or omitted characters. The MS spell-checker underlines all the unknown words, thus allowing the user to further inspect spelling errors which are out of reach for DIAC.

8. Conclusions

In comparing accuracies of two diacritics restoring systems, one has to take into account the processing unit (word or letter) and the way accuracy is defined because, otherwise the comparison might be very misleading. For instance, a very easy evaluation method starts with a regular text (containing all the required diacritics), strips off all the diacritics, run the automatic recovery and compares the original text with the text produced by the recovery algorithm. Non-identical units (words or characters) are considered mistakes. By dividing the number of mistakes to the total number of units in the text (words or characters) one gets what usually is reported as the error rate of the algorithm. This score depends on the considered unit, but also on the average character-length of a word (awg-length) and on average distance (awg-dist) between two consecutive correct diacritical characters. These two numbers may be language or even genre dependent. For languages where the average distance (avg-dist) between two correct diacritical characters is comparable or higher than the average length of a word (avg-length), as the is the case for Romanian and for many other languages, the evaluation of the character-based error rate looks always better the word-based error rate[4] (approximately avg-lenth better) .

As compared to our previous version (Tufis, Chitu, 1999), the present DIAC$^+$ implementation includes spelling corrector and processing for unknown words. It is more accurate due to the significant improvements in the underlying language model (the underlying lexicon is almost triple in size) as well as due to the increased accuracy of our tiered tagger. Also, in the previous version we used a combined language model (requiring the text to be re-tagged with each of the available language models and in the end combining the results (see (Tufis, 1999) for details). DIAC$^+$ is much faster because it uses a single tagging step, thus avoiding the time overhead of combined language model tagging (at a price of a less than 0.3% decrease of accuracy[5]). Since the coverage of the DIAC$^+$ essentially depends on the statistical underlying dictionary and the language model used by the tiered tagger, the system checks, on a regular basis, our linguistic web-service platform for newer language models and lexicons and updates itself accordingly.

The stand-alone version of DIAC$^+$ is implemented as a DLL and incorporates all the required information and processing tools. The web-service version does not have this problem, as the DIAC$^+$ code runs independently of MS-Office programs and thus, it is more appropriate for mass document processing than the DLL-based stand-alone version.

9. References

Bèze, M., Mérialdo, B., Rozeron, B., Serouault, A., M. (1994). Accentuation automatique de texte par des méthodes probabilistes. *Technique et sciences informatiques*, 13(6) pp. 797-815

Bobiceva, V. (2008). O altă metodă de restabilire a semnelor diacritice. In Pistol I., Cristea D. Tufiş D. (eds.): Resurse Lingvistice şi Instrumente pentru Prelucrarea Limbii Române, pp.179-188

Ceauşu, Al. (2006). Maximum Entropy Tiered Tagging, Janneke Huitink & Sophia Katrenko (editors), *Proceedings of the Eleventh ESSLLI Student Session*, ESSLLI 2006, pp. 173-179

Mihalcea, R. (2002). Diacritics Restoration: Learning from Letters versus Learning from Words. In Proceedings of CICLing, pp. 339-348.

De Pauw, G, Wagacha, P. W., de Schryver, G-M (2007). Automatic Diacritic Restoration for Resource-Scarce Language. In V. Matousek and P. Mautner (Eds.): TSD 2007, LNAI 4629, pp. 170–179

Simard, M. (1998). Automatic Insertion of Accents in French Texts. In Ide & Vuotilainen (eds) Proceedings of the Third Conference on Empirical Methods in Natural Language Processing , Granada, Spain, 27-35

Spriet T. and El-Bèze M. (1997). Réaccentuation automatique de textes. In *FRACTAL 97*, Besançon.

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufiş, D., Varga D. (2006). The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. *Proceedings of the 5th International Conference on Language Resources and Evaluation* (LREC'2006). Genoa, Italy, pp.2142-2147

Stolcke, A. (2002). SRILM - An Extensible Language Modeling Toolkit, in *Proc. Intl. Conf. Spoken Language Processing*, Denver, Colorado, September 2002

Tufiş, D., Chițu, A. (1999). Automatic Insertion of Diacritics in Romanian Texts. In *Proceedings of the 5th International Workshop on Computational Lexicography COMPLEX*, Pecs, Ungaria, 1999, pp. 185-194

Tufiş, D. (1999). Tiered Tagging and Combined Classifiers. In F. Jelinek, E. Nth (eds) *Text, Speech and Dialogue, Lecture Notes in Artificial Intelligence*, Springer, pp. 28-33

Tufiş, D. (2000). Using a Large Set of EAGLES-compliant Morpho-Syntactic Descriptors as a Tagset for Probabilistic Tagging, *International Conference on Language Resources and Evaluation LREC'2000*, Athens, 2000, pp. 1105-1112

Tufiş, D., Dragomirescu, L.(2004). Tiered Tagging Revisited. In Proceedings of the 4th LREC Conference, Lisabona, 2004, pp. 39-42

Zweigenbaum, P., Grabar, N. (2002). Accenting unknown words in a specialized language. In Proceedings of the Workshop on Natural Language Processing in the Biomedical Domain, ACL 2002 Philadelphia, July 2002, pp. 21-28.

Yarowsky, D. (1994). A Comparison of Corpus-based Techniques for Restoring Accents in Spanish and French Texts. In *Proceedings of the Second Annual Workshop on Very Large Corpora*, Kyoto, Japan

Wagacha, P.W., De Pauw, G. , Githinji, P. W. (2006). A Grapheme-Based Approach for Accent Restoration in Gĩkũyũ. In Proceedings of LREC2006, Genoa, Italy, pp. 1937–1940.

[4] An informal explanation: on average, each wrong diacritical character produces one wrong word (avg-dist > avg-length) so the nominators of the two scores are approximately the same; however the denominator of the character-based evaluation score is awg-length times larger than the denominator of the word-based evaluation score;

[5] Recall that not all tagging errors generate diacritics recovering errors: a 2% improvement of the tagging quality, has not a significant effect at diacritics restoration level.