Lexical Normalization of Spanish Tweets with Preprocessing Rules, Domain-Specific Edit Distances, and Language Models

Normalización léxica de tweets en español con reglas de preproceso, distancias de edición para dominio y modelos de lengua

Pablo Ruiz, Montse Cuadros and **Thierry Etchegoyhen**
Vicomtech-IK4
Mikeletegi Pasealekua 57
Parque Tecnológico de Gipuzkoa
20009 Donostia/San Sebastián
{pruiz,mcuadros,tetchegoyhen}@vicomtech.org

Abstract: We present a system to normalize Spanish tweets, which uses preprocessing rules, a domain-appropriate edit-distance model, and language models to select correction candidates based on context. The system's results at SEPLN 2013 Tweet-Norm task were above-average.
Keywords: Spanish microtext, lexical normalization, Twitter, edit distance, language model

Resumen: Presentamos un sistema de normalización de tweets en español, que usa reglas de preproceso, un modelo de distancias de edición adecuado al dominio y modelos de lengua para seleccionar candidatos de corrección según el contexto. El sistema obtuvo resultados superiores a la media en la tarea Tweet-Norm de SEPLN 2013.
Palabras clave: microtexto, español, castellano, normalización léxica, Twitter, distancia de edición, modelo de lengua

1 *Introduction and objectives*

Studies on the lexical normalization of Spanish microtext are scarce, e.g. Armenta et al. 2003, which predates Twitter and focuses on SMS. The lack of resources and tools for the normalization of Spanish tweets led us to develop a baseline system, identifying sources of error and means of improvement. The system comprises data resources to model the domain, as well as analysis modules.

Evaluating the system, we identified sources of error in dictionary and entity data, or in candidate-ranking elements like edit cost estimation and language model querying.

The Spanish tweet normalization system presented in this paper achieved above-average performance among a set of 13 competing systems at SEPLN's 2013 Tweet-Norm task[1].

The system's architecture and components are presented in Section 2, resources employed in Section 3, and settings and results-evaluation in Section 4. Conclusions and future work are discussed in Section 5.

2 *Architecture and components*

The system's architecture and components are shown in Figure 1 and explained in following.

2.1 Preprocessing

The preprocessing module was based on regexes and custom lists.

A set of case-insensitive regexes detected emoticons and delengthened OOVs with repeated characters, as well as mapping OOVs to DRAE[2] onomatopoeias. Repeated letters were reduced to a single letter, unless a word with a repeated letter was found in Aspell's Spanish inflected form dictionary (v1.11.3)[3]. E.g. *vinoo* was preprocessed to *vino*, but *creeeen* gives *creen*.

Custom lists were used to identify abbreviations, and expand them if needed. Lists

[1] www.sepln.org/?news=workshop-tweet-norm &lang=en

[2] Spanish Academy dictionary, www.rae.es

[3] `aspell -l es dump master | aspell -l es expand`

were also used to resegment tokens commonly written together in microtext. Microtext expressions such as RT or HT were considered in-vocabulary (IV).

2.2 Candidate generation

A minimum edit distance technique was used to obtain candidate corrections (Damerau, 1964). Up to two character edits (insertions, deletions, substitutions) were performed on preprocessed OOVs. Variants found in Aspell's dictionary were accepted as correction-candidates. The OOV itself was part of the candidate-set, since it's necessary to determine whether to keep the OOV (e.g. for proper nouns) or to edit it.

2.3 Candidate ranking

A candidate's rank prior to named-entity detection reflected the weighted combination of language model (LM) scores and edit distance.

For LM scores, the content of the n-gram looked up in the LM was configurable: either the trigram ending in the candidate, or a string with the candidate as the middle token, and up to nine tokens long. The LM score was the n-gram logprob returned by model lookup.

For distance scores, the Levenshtein distance between the OOV and each candidate was obtained. Each edit was assigned a cost. Costs were domain-specific, determined by surveying common errors in Spanish microtext. E.g. editing *k* as *q* (like in one of the edits from *kiero* to *quiero*) cost less than uncommon edits.

Costs were also inspired by spelling error frequencies for Spanish, reported by Ramirez and López, 2006. E.g. they found that 51.5% of errors were accent omissions. Accordingly, a cost model was created where replacing a non-accented character with its accented variant cost less than other substitutions.

Candidates where edit cost was higher than a threshold could be demoted or filtered out (configurable). A weighted sum of LM and distance scores was used to rank candidates.

Only the highest-ranked candidate moved forward to the next step in the workflow.

2.4 Entity detection

This step determines whether the highest-ranked candidate so far (as per LM and edit-distance scores) should be considered final, or if an entity-candidate should be proposed instead.

Exploiting NERC models from FreeLing 3.0[4] (Padró and Stanislovsky, 2012) was assessed. The models showed a strong tendency to label uppercase-initial tokens as entities. This was not optimal, since microtext does not reliably follow standard casing rules for proper-noun identification. FreeLing's confidence scores were also not a reliable indication of a token's proper vs. common noun status.

Given the difficulties in employing NERC, the following heuristics were implemented. For the best-ranked candidate according to the LM and edit distance scores, an uppercase-initial and an all-caps variant were created (e.g. *Messi* and *MESSI* for OOV *messi*). These variants were looked up in entity dictionaries. In case of a match, two factors determined whether to keep the non-entity candidate, or to accept an entity candidate. First, edit distance between the non-entity candidate and the original OOV. If larger than a threshold, the non-entity candidate was demoted. Second, segmental traits of the entity vs. non-entity candidate. E.g. if the entity contained sequences uncommon in Spanish (like *uu*), it was demoted. Accented characters, on the other hand, promoted either candidate.

2.5 Postprocessing

The original OOV's case could undergo decasing via regex application or candidate-set generation. The selected candidates were recased to match the original OOV's case.

Tokens were also uppercased when they followed a sentence delimiter.[5]

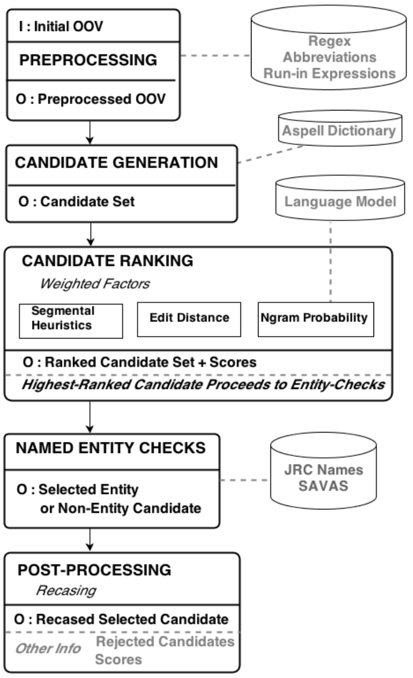

Figure 1: System Architecture

[4] nlp.lsi.upc.edu/freeling/

[5] The delimiters considered were **. ! ? " …**

3 *Resources*

As a known-word (IV) dictionary, DRAE was approximated by using Aspell's dictionary.

Entity lists were obtained from the JRC Names[6] database. A list of named entities manually annotated in the SAVAS corpus[7], which consists of 200 hours of Spanish news broadcasts from 2012 was also used. The SAVAS corpus was useful since it covers entities from current events, often discussed on Twitter. An entity-list for internal use at Vicomtech-IK4 was also employed.

Normalization does not require entity classification or linking, but merely identifying whether a token belongs to an entity or not. Accordingly, in our entity lists multiword entities were split into their tokens. Tokens for which a lowercase variant exists in Aspell's dictionary were filtered out.

For measuring candidate distance, a cost matrix was created. Costs were estimated by surveying the frequency of character substitutions in Spanish tweets, and inspired by (Ramírez and López 2006). Table 1 provides example costs. Using the table, editing *alli* to *allí* costs 0.5; *kiero* to *quiero* costs 1.5. Other cost models were also assessed (see Section 4).

Error	Correction	Cost (each)
a, e, i, o, u, n	á, é, í, ó, ú, ñ	0.5
k, *null*	q, u	0.75
p, a, z	m, u, k	1

Table 1: Edit Costs

We created three 5-gram case-sensitive language models with Kenlm[8], using an *unk* token. The corpora for the LMs contained one million sentences each, and vocabulary size was around 140,000. The *Subtitles* corpus contained film and documentary subtitles. The *Tweets* corpus contained Spanish tweets with no hashtags, usernames, or URLs. We also used a corpus extracted from *Europarl* (Koehn, 2005).

For the *Tweets* corpus, tweets with language value *es* and European time zones were collected in the spring of 2013. Only tweets for which Hunspell (v1.3.2) detected no errors were accepted. In order to decrease false positives, Hunspell dictionaries were enriched with entity lists. Tweet tokenization largely treated emoticons, URLs and repeated punctuation as single tokens. For tweets where there was at least 70% of token-overlap with other tweets, only one exemplar was accepted.

4 *Settings and evaluation*

Accuracy with the baseline settings (delivered for Tweet-Norm) was 0.610 for the dev corpus (475 tweets, 654 annotated OOVs) and 0.606 for the test corpus (564 tweets, 632 annotated OOVs). Edit distance and LM scores were weighted equally, and the LM had been trained on the *Subtitles* corpus (see Section 3).

Results with an LM trained on the *Tweets* corpus were comparable (0.599 for the test corpus and 0.594 for dev). Results with the *Europarl* LM dropped 5%. This suggests that LMs for tweet normalization can be trained correctly with off-domain texts containing short sentences and showing colloquial vocabulary and style, such as film subtitles.

The main sources of error for the baseline settings, based on a sample of 200 errors, are shown in Table 2.

Error Source	
List-and-Rule-Based Resources	**45%**
Subtotals	*%*
Regexes (for onomatopoeias etc.)	10.5
Gaps in domain dictionaries (Internet and social media slang)	9
Common missegmentations	8
Abbreviations	7
Known-words dictionary and Generic domain slang	6.5
Entity databases	4.5
Statistical Resources and Workflow	**30%**
Subtotals	*%*
Correction Model	12
Language Model and LM Queries	7
Entity Detection Heuristics	6
Ranking and Selection Criteria	5
Other Sources	**25%**

Table 2: System's Errors

Data problems account for 45% of errors. Adding abbreviations, and regexes for delengthening or mapping tokens to their DRAE form, would improve results. Other problems are gaps in dictionaries (e.g. missing Twitter slang, or DRAE word *pachanga* missing from Aspell), incorrect data (e.g. *bieber*

[6] optima.jrc.it/data/entities.gzip

[7] www.fp7-savas.eu/savas_project

[8] kheafield.com/code/kenlm/

in lowercase in JRC Names), or ambiguous data, such as Dutch name *Ruben* (from JRC Names) competing with the Spanish form *Rubén*. Adding a domain dictionary and accessing DRAE instead of Aspell would help solve these issues, as well as exploiting JRC Names' metadata to disallow foreign names in conflict with a Spanish name.

Issues with candidate scoring and ranking via the LM and distance metrics account for most of the remaining problems. *Correction Model* issues in Table 2 refer to cases where the distance score between the correct candidate and the OOV is too low to compete with candidates that have a better LM score in the context. The distance model in the baseline system is not context sensitive, e.g. replacing *k* with *q* and deleting *u* cost 0.75 each, regardless of context. We tested adding context sensitivity at the character level to the model. We defined a cost of 0.5 for corrections that involve two edits, but repair common errors in the domain, e.g. rendering *ki* as *qui*, *x* as *ch* or *wa* as *gua*. Distance-1 variants common in the domain that were given a cost of 0.5 are *p* for *pe*, *t* for *te* or *k* for *ka*, and others following the pattern of using a character as a shorthand for the sequence that sounds like the character's name. Corrections for colloquial generic variants like *ao* for *ado* were also given a cost of 0.5. Accuracy improved 1.06% in the test corpus and 0.61% in the dev corpus. The numbers are small, but no overcorrection occurred as a result of the modifications, which suggests a positive trend.

As for *LM* and *LM Querying* errors, the high proportion of OOVs in microtext makes it difficult to find relevant contexts to rank candidates. The following example illustrates a common pattern in the genre. The underlined tokens are OOV in <u>*@Idoia_LIA*</u> <u>*buenoa*</u> <u>*dias*</u> *mi vida*. Our system wrongly rendered *buenoa* as *buena* instead of *buenos*. The distance is 1 for both candidates. The token following OOV *buenoa* is OOV *dias*. Given that *buenos días* is a strong collocation, the corrected token, *días*, would be a useful feature for the LM to promote the correct candidate, *buenos*. However the tokens adjacent to the OOV under consideration (*buenoa*) are also OOV. Thus the LM scores are restricted to the unigram probability of candidates for the OOV to be normalized, without the benefit of context.

These difficulties to exploit LM information led us to test prioritizing distance scores over LM scores. By weighting distance scores at 70% and LM scores at 30% accuracy improved by 1.36% in the test set and by 0.76% in the development set.

Issues with *Ranking and Selection Criteria* in Table 2 are also related to the interaction between LM and distance scores. We implemented rules to modify the ranking for candidates that are missing an accent, but are otherwise exactly like the OOV. It is very likely for such candidates to be correct; accent-omissions cover 51.5% of the errors in Ramírez and López (2006). When candidates whose only error is an accent omission were ranked second, we required the difference between their LM score and the LM score of the first-ranked candidate to be higher than a threshold. By setting this threshold at 1.5, accuracy improved by 2.07% in the test set and by 0.76% in the development set.

Another way to improve candidate selection was the following: In the baseline, candidates at distance 2 were demoted, but allowed to compete with each other and with the OOV. This caused some overcorrection of IV tokens and of correct proper nouns, e.g. *hombrecillos* for *pobrecillos.* By filtering out candidates above a distance of 1.5, accuracy increased by 0.6% in the test set and 2.29% in the dev set.

5 Conclusions and future work

We presented a baseline system for the normalization of Spanish tweets. The system uses rules to preprocess OOVs into forms closer to IV tokens. Edit-distance scores and language model probabilities rank correction candidates. Edit-costs were estimated taking into account common errors in tweets. Candidate assessment via the language model posed difficulties, given the high frequency of OOVs in the candidates' context.

We also showed configurable settings that improve results by 5.09% (test corpus) and 4.42% (dev corpus).

In terms of future work, data problems caused 45% of the system's errors: additional preprocessing rules, IV entries, domain-specific entries like Twitter slang, or more accurate named entities would improve results

Context modeling is also an area to improve. Han and Baldwin (2011) devised a context model with features based on non-contiguous IV tokens, and used fuzzy matching between candidates and the context model.

References

Armenta, A., G. Escalada, J.M. Garrido, and M. A. Rodríguez. 2003. Desarrollo de un corrector ortográfico para aplicaciones de conversión texto-voz. *Procesamiento del Lenguaje Natural*, 31:65-72.

Damerau, F. 1964. A technique for computer correction of spelling errors. *Communications of the ACM*, 7(3): 171-176.

Han, B. and T. Baldwin. 2011. Lexical normalisation of short text messages: makn sens a #twitter. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Vol. 1: 368-378, Association for Computational Linguistics, Stroudsburg, PA, USA.

Heafield, K. 2011. KenLM: Faster and Smaller Language Model Queries. *Proceedings of the Sixth Workshop on Statistical Machine Translation*, 187-197. Edinburgh, Scotland, UK.

Koehn, P. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. *MT Summit 2005*.

Padró, L and E. Stanislovsky. 2012. FreeLing 3.0: Towards Wider Multilinguality. *Proceedings of the LREC 2012*. Istanbul, Turkey.

Ramírez, F. and E. López. 2006. Spelling Error Patterns in Spanish for Word Processing Applications. *Proceedings of LREC 2006*, 93-98.