Venetan to English Machine Translation: Issues and Possible Solutions

Suhel Jaber[1], Sara Tonelli[2], and Rodolfo Delmonte[1]

[1] Università Ca' Foscari, Venezia, Italy
[2] Fondazione Bruno Kessler, Trento, Italy

Abstract. In this paper we describe a prototype of a Venetan to English translation system developed under the *Stilven* project financed by the Regional Authorities of Veneto Region in Italy. The general approach is a statistical one with some preprocessing operations both at training and translation time (ortographic normalization and POS tagging to make use of factored models) which are needed especially to overcome two main problems: the scarcity of Venetan resources (our Venetan-English corpus is made up of only 13,000 sentences, amounting to 128,000 Venetan tokens) and the diasystemic nature of Venetan, which really represents an ensemble of varieties rather than a single dialect. We will present in detail the problems related to Venetan, our ideas to solve them, their implementation and the results obtained so far.

Keywords: Machine translation, less-resourced languages, language varieties

1 Introduction

Stilven[3] [7] is a project approved in December 2007 which started its activities in February of the following year. The task was creating a computational infrastructure for the analysis and translation of Venetan language (see for example [18]). Venetan is a dialect nowadays but was the official language of the Veneto Republic for as long as 8 centuries, up to the end of the XIXth century, when the Republic became part of newborn Italian nation. Since then, Venetan has been slowly abandoned in favour of Italian. Nowadays, depending on the region, Italian speakers can usually master a dialect and the main language. In particular, Venetan speakers show a much wider usage of dialect - their original language - in most working places, in the family and in social life.

Venetan proficiency by local speakers has been lately assessed as reaching 75% of the population in the Veneto region. Furthermore, more than 5 million speakers are scattered around the world, since in the past two centuries a large part of the population emigrated from Italy to other countries. Also, a small community of Venetan speakers is very active on the Internet, contributing to the diffusion of this language through several web-sites including a version of Wikipedia in Venetan (`http://vec.wikipedia.org/wiki/Vèneto`).

[3] `http://project.cgm.unive.it/stilven_en.html`

Venetan dialect is now considered a *diasystem*, where speakers use their own variety and manage to understand each other. Venetan is nowadays a spoken dialect, which has developed a number of varieties. In [6][4] , seven example dialogues are reported, each corresponding to a variety spoken in a Venetan city (i.e. Venice, Vicenza, Rovigo, Padua, Treviso, Belluno and Verona).

As to similarities, all varieties apart from Venetian use subject clitic inversion in questions. As an example of syntactic differences, we mention that Belunese is the only variety to allow verb fronting before question word: 'Féu che' (Do what), and clitic subject for weather verbs, 'Piòvelo' (It rains). Lexical differences are many and constitute the main distinguishing element: for example 'céo' (boy), is only used by Trevisan, while 'sani' (see you) is only used by Belunese. As to the remaining differences, they are all understood by the majority of Venetan people.

2 *Stilven* Project Objectives and Activities

Very much like what has been done with METIS [5], our system aims at translating free text input by taking advantage of a combination of statistical, pattern-matching and rule-based methods. The following goals and premises were defined for the project:

- use simple NLP tools and resources,
- use bilingual hand-made dictionaries,
- use Italian as intermediate language,
- use translation units at sentence boundaries,
- use different tagsets for source language (SL) and target language (TL).

Moreover, a translation system that has to cope with varieties has two main problems to solve:

- lexicon extension including all specialized items present in one variety and not in the others;
- grammatical flexibility that must properly process sentences with different structural organization according to each variety.

Syntactic peculiarities will be discussed in the next sections, whereas the problem of accounting for lexical varieties has been tackled by implementing a number of different lexica which refer at the same time to the four main varieties, to Italian and to English.

3 Linguistic Resources and Orthographic Normalization

In parallel to the implementation of the *Stilven* system, several linguistic resources were created in order to support the development of NLP applications

[4] Available at `http://www.linguaveneta.it/sussidiario.html`

for Venetan. First, we collected as much text as possible from the web and from people collaborating on a voluntary basis. Texts collected were then homogenized as to the orthography. They are organized into 7 different genres and include children stories, the translation of a book of American history, the translation of 'The Little Prince' by Antoine de Saint-Exupéry, the translation of a series of political newspaper articles, the translation of famous quotes taken from the LOGOS website (`www.logos.it`), the translation of a manual of the Venetan orthography rules [8] and a small set of especially built sentences directed to grammatical issues. As a whole, we collected texts for 200,000 tokens.

Also, frequency lists were compiled based on these texts. The lists were then the basis for the wordform lexicon of Venetan, which has been compiled on the basis of the Italian one available in our laboratory, thus comprising in each entry the corresponding Italian wordform and lemma. Semantic and syntactic properties of the Venetan wordform would then be derived directly from the Italian fully specified subcategorized lexicon.

We then normalized a big translation lexicon (52,000 entries) containing lemmas of Venetan paired with Italian and English. Moreover, we used parallel English-Italian texts to derive multiwords that could then be matched with those present in the Venetan-English parallel texts. From these materials we managed to collect a small dictionary of 200 multiwords which include very frequent function multiwords, like adverbial and prepositional locutions.

Normalization is a common issue to many languages in the world such as Arabic, Chinese and Japanese, which share the same problem of orthographic variation. Normalization is needed to allow the wordform to be checked against a lexicon where standardized orthography has been used. In our case, lexemes are produced in the lexicon with an official orthography according to the GVU (Unified Venetan Writing) obeying rules formulated some years ago by linguists [8] and published in the website of Veneto Region[5].

To make a comparison with Arabic, we see that orthographic variations may arise for a number of reasons, the first of which is certainly the dialectal variation. Then there is the objective problem of rendering some phonemes into a romanized valid corresponding character. As a result, an Arabic name may have hundreds if not thousands different variants in its romanized version. Coming back to Venetan, the problem is not so acute and the solution that can be adopted is the one that is also applied to other languages, that is an orthographic rule-based approach. In other words, due to the small number of variants it is not fit to use a lexicalized approach where all variants are stored after being automatically and then manually validated, for instance on the basis of their frequency of occurrence on the web. It will then be sufficient to list all cases of orthographic variations occurring in Venetan and then to formulate a corresponding set of rules. These rules coincide with what has been done for Arabic, for instance. In particular, consider the following rule for the recognition of some typical characters. As may be seen, the starting point is the corresponding phoneme, and on

[5] `http://win.elgalepin.org/gvu/index.html`

the right hand side there is a list of possible graphemes. Note that the mapping is one-to-many.

Example 1.
/dz, ts/ → d dh t z th

/k/ → k q c ch

/j/ → j g dj

The other remarkable orthographic problem concerns the need to use word stress on E and O to differentiate open vs. closed phoneme. The difference is crucial to characterize minimal pairs which otherwise would not be disambiguated, as for example in *béco* (goat) vs. *bèco* (beak), *péxo* (weight) vs. *pèxo* (worse), *bóte* (keg) vs. *bòte* (strikes), *fóla* (crowd) vs. *fòla* (lie).

So here again the problem lies in the lack of native speakers' awareness of the need to introduce such diacritics because they do not hear the ambiguity. Normalizing in this case is more complex because the meaning changes according to the type of accent chosen.

4 The Tagger

The first tool we worked on was the tagger of Venetan based on a semi-automatically annotated corpus of 128,000 words.

To increase the entries of the training corpus, we decided to decompose all idiomatic expressions and all locutions which amounted to some 2,000 entries. We also intended to use the 52,000 lexical entries that we collected as described in Section 3. So we added an article in front of all nouns and adjectives. Then, we composed pseudo sentences by joining nouns and adjectives to infinitival verbs (available in the lexicon) and adverbs. In this way, we collected another additional 85,000 entries which increased the size of the training corpus to almost 200,000 tokens.

4.1 The tagset

One of the interesting aspects of this work was the tagset we eventually came up with after a number of dubious cases, on the basis of our previous work on Italian. The POS with the corresponding meaning are reported in Table 1.

The most interesting cases regard the subdivision of cliticized verbs into three subcategories, namely VCL (verb cliticized, not inflected), VCLI (verb cliticized, inflected) and VPRON (verb inflected with cliticized subject pronoun). The reason for this subdivision is due to the need to separate VPRON from VCLI. While this is not needed in Italian and other Romance languages, Venetan requires another class of cliticized verbs, because it allows subject pronouns in questions. Here, the peculiarity is not only constituted by the amalgam in a

POS	Extended POS	POS	Extended POS
abbr	abbreviations, acronyms	num	number
ag	adjectives	par	parenthetical - punctuation “" ( ) -
art	articles - definite e indefinite	pk	complementizer “che"/that
avv	adverbials	prep	preposition
clit	clitic generic	part	preposition amalgamated with article
clitg	clitic “ghe” / there, you	pavv	prep./adverb “con su sora soto” /
nt	noun temporal		with, on, over, down
clits	clitic “se” / reflexive,impersonal	poss	possessive
ccom	conjunction “come" / like	prog	progressive periphrastic “drio”
cong	conjunctions “or, and"	pron	pronoun personal
congf	conjunction sentential	q	quantifier
cosu	conjunction subordinate	rel	relative pronoun “che" /
neg	negation		that, which, who, whom
date	number date	relob	relative pronoun oblique “cui” / whose
deit	deictic pronoun	relin	relative pronoun indefinite
dim	demostrative adjective	sect	sector number followed by fullstop
np	noun proper geographic		or parenthesis
dot	punctuation	v	verb inflected
fw	foreign word, also non-words	vav	verb “ver” / to have auxiliary and lexical
in	intensifier	vcl	verb cliticized non inflected
ind	indefinite quantifiers	vcli	verb cliticized inflected
int	interrogative pronouns	vd	verb gerundive
intj	interjections	vi	verb infinitival
n	noun common	vprog	verb progressive “star" / to stay
nh	noun proper human, appellation, social role	vpron	verb inflected with
punt	punctuation, : ;		cliticized subject pronoun
punto	punctuation sentence end . ? !		

Table 1. POS Tagset and explanation

question, but it is the clitic form that is very special. Final vowel is usually ‘-o’ for ‘-to’ (you) modifying the normal ‘-ti’ ending with ‘-i’. The use of ‘-ti’ is present and is determined by a phonological rule: the presence of a nasal ‘-n’ in the verb ending, as in ‘gonti’ (have you), ‘fonti’ (make you), ‘sonti’ (are you). We counted 647 cases of VPRON, 341 cases of VCLI and 1214 cases of VCL. It is important to note that these forms are very productive in conversations.

4.2 Tagger comparison

Given the tagged training corpus containing 218,864 tokens, we decided to compare the performance of two supervised taggers: the *Brill's Tagger* [4] in the Python implementation included in the NLTK suite [1][6], and the *HunPos Tagger* [9] [7], an open source reimplementation of the well-known *TnT tagger* [3],

[6] Available at `http://www.nltk.org/`

[7] Available at `http://code.google.com/p/hunpos/`

based on HMM. In this way, we compare for the first time the behaviour of Brill's transformation-based approach and of HMM-based statistical approach on Venetan documents. A similar comparison was performed for example by [10] for Bangla, by [16] for Dutch and by [2] for English.

Brill's tagger relies on a *transformation-based approach*, which combines a rule-based approach and statistical methods. In short, it picks the most likely tag based on a training corpus and then applies a certain set of rules to see whether the tag should be changed to anything else. Then, it saves any new rules that it has learnt in the process, for future use. In this way, Brill's tagger tries to transform an initial bad tagging into a better one in an iterative fashion.

As for stochastic taggers based on HMM (Hidden Markov Models), the training set is used to compute a statistical model that, given a word sequence, chooses the tag sequence with maximum probability.

The tagger implementation we use, called *HunPos*, is based on second-order Markov model, and the output probability is based on the previous tag in addition to the current tag. It also includes a suffix guessing algorithm to deal with unknown words.

As reported by [2], the performance of the HunPos tagger on WSJ data, measured by its error-rate, proved to be much better than that of the Brill's tagger if a small training-set is used. For large training-sets of 100,000 sentences the performances seem to be about the same, with the former tagger edging out ahead on the larger tagsets and the Brill's tagger edging ahead on the small tagset. However, the advantage of Brill's tagger is that it is easier for the user to manually correct the automatically induced knowledge of the tagger.

While the documents used for training are a collection of texts coming from the different sources reported in Section 1, the test set includes 371 sentences (10,493 tokens) translated from scientific articles in the domain of biology. In this way, we make the test more challenging because we introduce also a domain shift.

Since our main goal is to understand which of the two taggers performs better in order to integrate it into the machine translation system in a future step, we focus our evaluation on the tokens which got a different annotation from the two taggers. Results are reported in Table 2.

N. of tokens with different annotations	2052
N. of correct labels assigned by HunPos	1517
N. of correct labels assigned by Brill	365
N. of wrong annotations by both taggers	170

Table 2. Evaluation of taggers performance

Our evaluation confirms the results obtained for English [2]: also for Venetan the *HunPos* tagger performs remarkably better than the *Brill's tagger*. In particular, 74% of the diverging tags are correctly labeled by *HunPos* while only

18% of them are correct assignments by the *Brill's tagger*. The latter assigns the N label, which is the most frequent one in the training set, to unknown cases, while in *HunPos* the guesser seems to work particularly well, detecting also many foreign words, proper names, etc. Most of the cases in which both taggers fail concerns the classification of clitics, which are often homograph of articles, prepositions and pronouns, and can occur in different positions inside the sentence. Therefore, their recognition is one of the main challenging tasks of Venetan tagging.

We also perform a standard ten-fold cross-validation in order to assess the overall performance of *HunPos* on in-domain data. The final accuracy amounts to 90%, which is below the performance of state-of-the-art taggers for other languages, but is still a promising result given that the training corpus was quite small and it was enriched with automatically-generated pseudo sentences.

5 Venetan to English Translation

Problems related to Venetan translation into English and viceversa are very close to those encountered when translating from/into Italian. The most interesting types of problems include subject clitic doubling, amalgams (prepositions + article; verb + enclitic), proper nouns preceded by articles and subjects adjoined as enclitics in interrogative sentences.

To implement our Venetan-English machine translation system we have decided to use a statistical approach [12]. Unlike rule-based approaches, statistical machine translation allows for the automatic induction of a phrase dictionary based upon sentence-aligned corpora: given these corpora, available algorithms like the one implemented in the GIZA++ package [15] are able to infer probabilities of alignments between source and target single words or phrases (where the term *phrase* indicates merely a sequence of words and has no linguistic connotation whatsoever) and build a so-called *translation model*.

The probabilities contained in these phrase-tables for each entry are not the only factor affecting the overall probability of a potential translation over another. There is also a measure of how natural a potential target sentence is, approximated via n-grams probabilities extracted from a monolingual corpus in the target language: the result of this procedure constitutes the so called *language model*. Finally, a *reordering model* accounts for word displacement phenomena. These probabilities contribute to the overall probability of a certain target sentence being the translation of a given source one according to automatically computed weights. The toolkit we have used to implement our system is the Moses open source toolkit [14].

5.1 Reasons for the choice of a statistical approach

Going for a statistical approach to machine translation allows for the possibility to automatically learn the bilingual dictionary by training the translation model. Aside from the fact that coding all the rules by hand in a rule-based approach would prove much longer a task than automatically inferring a translation model, the real problem is that to actually code all the possible rules, we should make available a sufficiently large corpus of the source language which should undergo analysis first, and in the case of Venetan, only thinking of morphologically analyzing the plethora of irregular words turns out to be an extremely complicate matter. Venetan allows for the cliticization of subject pronouns, a feature rare in Romance languages. The problem arises when the stem to which the subject pronoun is attached, i.e. the verb, undergoes changes which have not been thoroughly studied yet and for which there does not appear to be a constant rule. Here is an example[8]:

1-vuto 2-magnar 3-con 4-mi 5-?
1-do you want 2-to eat 3-with 4-me 5-?

In the Venetan, sentence the token 'vuto' should be morphologically analyzed as verb stem 'vu-' and subject pronoun clitic '-to', as opposed to its non contracted form 'ti vol'. Now coding generalized rules to transform 'vu-' into 'vol' (or vice versa) proved to be a hard task because other verbs behave in a different way, for instance 'gheto' for 'ti ga ' in the following sentence:

1-'sa 2-gheto 3-da dir 4-?
1-what 2-do you have 3-to say 4-?

By looking at these examples it becomes clear that given our current knowledge of Venetan, the best way to deal with such phenomena is that of a direct mapping between the full-fledged verb form and its contracted stem version via ad hoc rules (i.e. a dictionary lookup pass), but again, that defeats the purpose of manually coding the dictionaries. So we decided it would be easier to just rely on alignment algorithms and go with the statistical approach. Note that the English translation of both examples above is what our system currently outputs, which is indeed the correct translation.

5.2 Issues encountered in using the statistical approach for an under-resourced language

Statistical machine translation, being inherently a data-driven approach, works well when there is lots of data. Given that our corpus is so small (i.e. 13,000 parallel sentences with 128,000 words), we have encountered some problems. As far as parameter tuning is concerned, for example, what we have found out is that the language model weight has to be lowered from the default provided by the toolkit to optimize our results. Even in cases where correct alignments are actually inferred during the translation model training phase, there still can arise problems. Namely, the probabilities assigned to correct alignments are not generally very conclusive, and therefore the decisive discriminant for the choice

[8] In all examples, spacing and numbering render graphically segmentation of the source sentence into source phrases and selection of best target phrase for each source one as performed by the decoding algorithm during translation

of a translation over another turns out to be the language model; this can hurt translation quality in some cases, an example of which is reported below.

1-clifford 2-xe ndà 3-via
1-go 2-went 3-away

The source sentence contains a proper noun ('clifford'), the frequency of which is probably too low in the Europarl corpus [11] used to the train the language model with the toolkit by [17]. If the language model can actually make a difference, it will steer the decision of translation towards the token most frequently found in that position, and since 'clifford' does not appear often enough, it will consider even bad sequences of known tokens a better choice. In formal terms, there certainly existed a trigram such as '<null> <null> go' in the language model, where <null> in the n-gram world is the null token that is inserted n-1 times at the beginning of a sentence to evaluate the probability for that very sentence of beginning with some other non-null token.By lowering the language model weight, the proper noun (which indeed appeared several times in the training corpora used for the translation model) finally makes it to the target sentence, but the consequences of diminishing the language model's impact are reflected in crunchier overall translations, as the following one. Note that the correct English translation would be 'clifford went away'.

1-clifford 2-xe 3-ndà 4-via .
1-clifford 2-is 3-to go to 4-away .

Increasing the frequency of proper nouns like 'clifford' in our n-grams by collecting ad-hoc data is not a proper solution, as it would create a language model that is less representative of the reality we are trying to model.

5.3 Unfactored vs. factored models

Another advantage of state-of-the-art statistical methods nowadays is that linguistic knowledge can easily be integrated to enhance translation quality. More specifically, factored translation models [13] allow for the representation of a token in the sentence-aligned corpora as a vector of factors and for the learning of mappings between phrases of one (or more) particular source token factors to phrases of one (or more) particular target token factors, that is to say a training phase in which the alignment occurs between phrases of (sub)vectors of factors. In order to have a preliminary idea of the impact of factored translation models on Venetan, we did some initial tests with manually or semi-automatically annotated tags. Once the pipeline is consolidated, we will also integrate the *HunPos* tagger presented in Section 4.

The main reason for resorting to factored translation models in our case has been the possibility of learning alignments between phrases of source vectors made up of a wordform plus its relevant tag and phrases of target wordforms. In other words, our setup implements a single translation step in which the input factors are surface form and part of speech in Venetan, and the output factor is surface form in English. In this way we have been able to disambiguate the plethora of Venetan homographs, as in the sentence below, where '|' is the factor separator. The correct English translation would be 'His/her father is good'.

```
1-so|poss pare|nh 2-xe|vex 3-bon|ag 4-.|punt
1-his father make 2-is     3-good   4-.
```

This is still not perfect (the word "make" has been wrongly added) but surely better than the unfactored version below.

```
1-so 2-pare  3-xe   4-bon .
1-i  2-think 3-it is 4-good .
```

The homograph 'pare' actually means both 'father' and 'seems', and the unfactored version favours the verb meaning, whereas the source sentence clearly means 'his father is good'. The factored model gets the meaning part better but shows evidence that the word alignment algorithm failed in segmenting phrases in the target language corpus at translation model training time.

5.4 Other shortcomings of the statistical approach for under-resourced languages

Even if a given vector appears in several sentences of the source language corpus, if it is not repeatedly translated into the same word in the aligned target corpus sentences, the algorithm cannot infer an alignment that spans only that single source vector. So what it really does is to singly align only the words that appear more often, and consider the other intervening words in the sentence as one big phrase that gets aligned in its entirety to what remains of the target sentence. Therefore, if a sentence containing one of those problematic words is inputted for decoding, it can be correctly translated only if that word is part of a sequence that is identical to the one to which it belonged in the training set, otherwise the problematic word will simply pass onto the output untranslated. Below you can see examples of this behaviour. Note that the English output is correct.

```
1-se|cosu 2-no|neg no|neg podaria|vsup nar|vi pi|q coi|part pàtini|n
1-if      2-i couldn't skate anymore
```

This is a sentence taken directly from the original corpus. A correct segmentation here should have separated 'se|cosu no|neg' from the rest of the source sentence and should have mapped its translation to 'then', which is what indeed appears in the original target sentence instead, rather than 'if'. Yet 'if' is a correct translation of 'se|cosu' by itself, and since this source-target pair has been noticed by the learning algorithm many times in the sentences of the original sentence-aligned corpora, a potential mapping for it has been established with a degree of confidence high enough to authorize its separate translation at decoding time. The rest of the sentence, on the other hand, is translated as a single phrase, and exactly the way it was found in the target corpus sentence during training time. To stress this point, we show how in the example below, the vector ‘pàtini|n’ is not translated anymore just because we interrupt the long phrase found in the original corpus with a ‘,|punt’.

<table>
<tr><td>1-se|cosu no|neg</td><td>2-,|punt</td><td>3-no|neg</td><td>4-podaria|vsup</td><td>5-nar|vi</td><td>6-pi|q</td><td>7-coi|part</td><td>8-pàtini|n</td></tr>
<tr><td>1-if not</td><td>2-,</td><td>3-i</td><td>4-could</td><td>5-go</td><td>6-more</td><td>7-with</td><td>8-pàtini</td></tr>
</table>

Finally, we would like to point out that there are no reordering problems with our translations.

6 Conclusions and future work

In this paper we have presented some issues related to the development of a Venetan to English machine translation system. Since Venetan is a diasystem, many challenges have to be tackled while creating NLP tools, from the poor resources available to orthographic normalization.

While detailing the applications implemented so far, we have suggested some solutions to the above mentioned problems. We have further described the tools under development, including a PoS tagger and a statistical machine translation system. Since we do not have large enough Venetan-English corpora to overcome wrong or missing alignment problems, and resources are generally scarce for Venetan, we will try to exploit Italian also. We will not use Italian as a pivot language within the statistical system as that would not really solve the problem, since Italian-Venetan language resources are as scarce as English-Venetan ones and therefore using the pivot language would only worsen translation results. What we will do instead is to transform an Italian text contained in a large En-Ita parallel corpus into Venetan text with a rule-based approach. The idea is that manually coding rules to translate from Italian into Venetan is still less expensive than coding rules to transform Venetan into English, as Venetan is in fact a dialect of Italian, with which it shares a lot of grammatical rules and of lexical entries. In addition, dialects lack native words for a number of lexical domains, like for instance, bureaucratic domain, scientific domain etc, which is where we will look into next.

Finally, it must be said that we have carried out the pre-processing of most of our language resources in a semi-automatic way with the support of human translators, annotators, editors, etc., and that we have used the whole amount of the resulting parallel corpus for training our system. The lack of a numerical evaluation of its performance stems from the problems we have encountered in automatically normalizing, categorising, and tagging existing (out of domain) Venetan texts, such as those contained in the Venetan version of Wikipedia, which we would like to use for the evaluation of our translation system. This is another issue we are working on.

References

1. Bird, S., Loper, E.: NLTK: The Natural Language Toolkit. In: Proceedings of the ACL demonstration session. pp. 214–217. Barcelona, Spain (2004)
2. Block, S.: A Comparison of three part-of-speech taggers. Master's thesis, Uppsala Universitet (2009)
3. Brants, T.: TnT - A Statistical Part-of-Speech Tagger. In: Proceedings of the Sixth Applied Natural Language Processing Conference ANLP-2000. Seattle, WA (2000)
4. Brill, E.: Unsupervised Learning of Disambiguation Rules for Part of Speech Tagging. In: Proceedings of the Workshop on Natural Language Processing Using very Large Corpora. Boston, MA, USA (1997)
5. Carl, M., Melero, M., Badia, T., Vandeghinste, V., Dirix, P., Schuurman, I., Markantonatou, S., Sofianopoulos, S., Vassiliou, M., Yannoutsou, O.: METIS-II: Low resource machine translation. Machine Translation 22(1-2) (2008)
6. Cortelazzo, M.: Noi Veneti - Viaggi nella storia e nella cultura veneta... Regione Veneto (2001)
7. Delmonte, R., Bristot, A., Tonelli, S., Pianta, E.: English / Veneto Resource Poor Machine Translation with STILVEN. In: BULAG 33 - International Symposium on Data Sense Mining, Machine Translation and Controlled Languages, and their application to emergencies and safety critical domains (2009)
8. Giunta Regionale del Veneto: Manuale di Grafia Veneta (1995), available online at http://www.veneto.org/gvu/
9. Halácsy, P., Kornai, A., Oravecz, C.: HunPos - an open source trigram tagger. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, Demo and Poster Sessions. pp. 209–212. Prague, Czech Republic (2007)
10. Hasan, F., UzZaman, N., Khan, M.: Comparison of different POS Tagging Techniques (n-gram, HMM and Brill's tagger) for Bangla. Advances and Innovations in Systems, Computing Sciences and Software Engineering pp. 121–126 (2007)
11. Koehn, P.: Europarl: A Parallel Corpus for Statistical Machine Translation. In: Proceedings of MT Summit (2005)
12. Koehn, P.: Statistical Machine Translations. Cambridge University Press (2010)
13. Koehn, P., Hoang, H.: Factored Translation Models. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Prague, Czech Republic (2007)
14. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: Open Source Toolkit for Statistical Machine Translation. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics. Prague, Czech Republic (2007)
15. Och, F.J., Ney, H.: A Systematic Comparison of Various Statistical Alignment Models. Computational Linguistics 29(1), 19–51 (2003)
16. Stehouwer, J.H.: Comparing a TBL Tagger with an HMM Tagger: Time Efficiency, Accuracy, Unknown Words (2006), internship report
17. Stolcke, A.: SRLIM - An Extensible Language Modeling Toolkit. In: Proceedings of the International Conference on Spoken Language Processing. Denver, USA (2002)
18. Tonelli, S., Pianta, E., Delmonte, R., Brunelli, M.: VenPro: A Morphological Analyzer for Venetan. In: Proceedings of the 7^{th} International Conference on Language Resources and Evaluation. Valletta, Malta (2010)