METIS: STATISTICAL MACHINE TRANSLATION USING MONOLINGUAL CORPORA

Nikos Ioannou INSTITUTE FOR LANGUAGE AND SPEECH PROCESSING

Abstract

Το METIS[1] είναι ένα πρωτότυπο σύστημα μηχανικής μετάφρασης το οποίο χρησιμοποιεί τεχνικές αναγνώρισης προτύπων για να ανασύρει μεταφράσεις προτάσεων της γλώσσας πηγής από ένα μεγάλο σώμα κειμένων της γλώσσας στόχου. Η κύρια πρωτοτυπία το συστήματος είναι ότι δεν χρησιμοποιεί δίγλωσσα σώματα κειμένων (bitexts). Ταυτόχρονα, είναι εύκολα προσαρμόσιμο σε οποιοδήποτε γλωσσικό ζεύγος, με την προϋπόθεση ότι υπάρχει ένα μεγάλο σώμα κειμένων της γλώσσας στόχου, ένα δίγλωσσο λεξικό (ένα για κάθε γλωσσικό ζεύγος) και ένα μικρό σύνολο κανόνων που αντιστοιχίζουν τεμάχια της δομής της γλώσσας πηγής σε τεμάχια της δομής της γλώσσας στόχου.

Keywords

statistical, example based, machine translation, monolingual corpora, METIS.

1. Aims – General characteristics of the system

The METIS system aims at obtaining free text translations of reasonably high quality from **large grammatically annotated and lemmatised monolingual corpora with pattern-matching techniques**. Currently, all Statistical Machine Translation (SMT)/ Example Based Machine Translation (EBMT) approaches are based on very large parallel and aligned corpora or *bitexts*, while rule-based systems require complicated grammars which consists of detailed lexical and large sets of rules. The main novelty introduced by METIS is the elimination of the use of *bitexts* and this is achieved by using pattern matching techniques and by adopting a small set of hand-crafted linguistic rules. There are, thus, two pillars, which the system rests upon: (i) the pattern-matching mechanism and (ii) the sentence comparison mechanism.

[1]METIS was funded by the EU as an IST-FET project (METIS-IST-2001-32775).

2. Input

The system′s requirements in terms of resources are the following

- Bilingual lexicon file with PoS information
- Tag-mapping Rules file
- Tagged and Lemmatized source language sentence
- Tagged and Lemmatized target language corpus
- Comparison Weights file.

Each type of the resources will be described in the sections that follow.

2.1 Bilingual Lexicon

The lexicon provides two types of input: (i) simple word entries (lemmata) and (ii) phrases that cannot be translated word-by-word in the target language (lexical phrases). In addition to lexical phrases and lemmata translations, part of speech information is also provided in the lexicon (table 1):

κόβω#ταχύτητα	Vb#*	slow#down	V??#*
…	…	…	…
αναπνοή	No	breath	N??
ανηφόρα	No	uphill	N??
από	As	by	PRP
από	As	over	PRP
από	As	since	CJS
…	…	…	…

Table 1 *– Example part of the bilingual lexicon*

Lexical ambiguity is encoded in the lexicon by listing entries, eg. there are 3 lines (3 entries) for the Greek preposition "από" , on table 2. No part-of-speech information is provided for indeclinable words within phrases (the corresponding field in the second and fourth column is annotated with an asterisk) and the system tries to match the exact token.

2.2 Tag Mapping Rules

The bilingual lexicon is applied to the lemmatized source language string. A string is generated consisting of all the possible target language lemmata corresponding to the lemmata in the source language string and occurring in the order imposed by the source language string. Tag mapping rules transform the (rudimentary) grammatical structure of the output string to a structure that is closer to that of the target language sentence(s) sought. The form of the rules is shown in Fig. 1. The source language patterns occur on the left-hand side of the rule while on the right hand side of the rule, the changes that must be incorporated in the output string are shown. The structure is described as a regular expression of lemma-tag patterns:

```
[1\ \ VbMnIdPr______IpAv__] ->        [1\ \ VVB-VVZ] |
                                      [\ \ VBB-VBZ][1\ \ VVG]|
                                      [\ \ VHB-VHZ][\ \ VBN][1\ \ VVG] ;
[1\ *\ AtDf*][\ *\ *]{,2}[2\ *\ NoPr*] -> [2\ \ NP0];
```

Figure 1 *– Tag mapping rules*

A full description of the rules functionality is given in Section 3 (System Description).

2.3 Tagged and Lemmatized Source Language Sentence

Source language sentences are tagged and lemmatized. For every word in the source language file, the system expects to find three entries, which encode the following information: the word (or token), the lemma of the word and the tag with morphological information. Table 2 contains an example sentence with all the information necessary for the system. Greek sentences were tagged using the ILSP-Parole tagset.

Word (Token)	**Lemma**	**Tag**
Η	ο	AtDfFeSgAc
γυναίκα	γυναίκα	NoCmFeSgNm
καθαρίζει	καθαρίζω	VbMnIdPr03SgXxIpAvXx
το	ο	AtDfNeSgAc
μήλο	μήλο	NoCmNeSgAc
.	.	PTERM

Table 2 *– Source language sentence form*

2.4 Tagged and Lemmatized target language corpus

METIS requires a target language corpus that has been annotated in the same way as the source language sentence (described in Section 2.1). An example of an annotated target language sentence is given in table 3. The sentences in the target (English) language corpus were tagged using the CLAWS5 tagset.

Word(Token)	**Tag**	**Lemma**
Prosperity	NN1-NP0	prosperity
is	VBZ	be
not	XX0	not
only	AV0	only
related	VVN	relate
to	PRP	to
the	AT0	the
number	NN1	number
of	PRF	of
cars	NN2	car
per	PRP	per
inhabitant	NN1	inhabitant
.	PUN	.

Table 3 – Target language sentence form

2.5 Weights Comparison File

The weights comparison file contains the set of parameters used for comparing the transformed source language sentence and the sentences that are picked from the corpus. The functionality of the weights file will be described in detail in the system description section (Section 3). An example weights file is given in Fig. 2:

```
<BasicSettings>
	PadCost=60 ;
	DistancePenalty=100;
	RequiredDistance=3;
	BasicRange=[-200,200];
	SentenceWindowing=ON; %Possible values ON and OFF%
	DistanceLog=ON;
	SpeedUp=OFF;%Possible values ON and OFF (This feature is still inactive)%
<\ BasicSettings>
<TagPattern VB*>
	ocf=0.5 ;
	tcf=0.1 ;
<\ TagPattern>
…
<TagPattern AT*>
	ocf=0.1;
	tcf=0.5 ;
<\ TagPattern>
…
```

***Figure 2** – Example weights file*

3. System Description

METIS uses a large target language corpus and pattern matching techniques in order to find those target sentences in the target language corpus that provide the optimal translation of the given source-language sentence. Thus in the current development state, METIS has the capacity of a translation memory.

The system is divided in three major parts:

1. Lemma-to-Lemma translation
2. Rule application for structural transformation
3. Corpus search for optimal translation

Figure 3 demonstrates the system functionality.

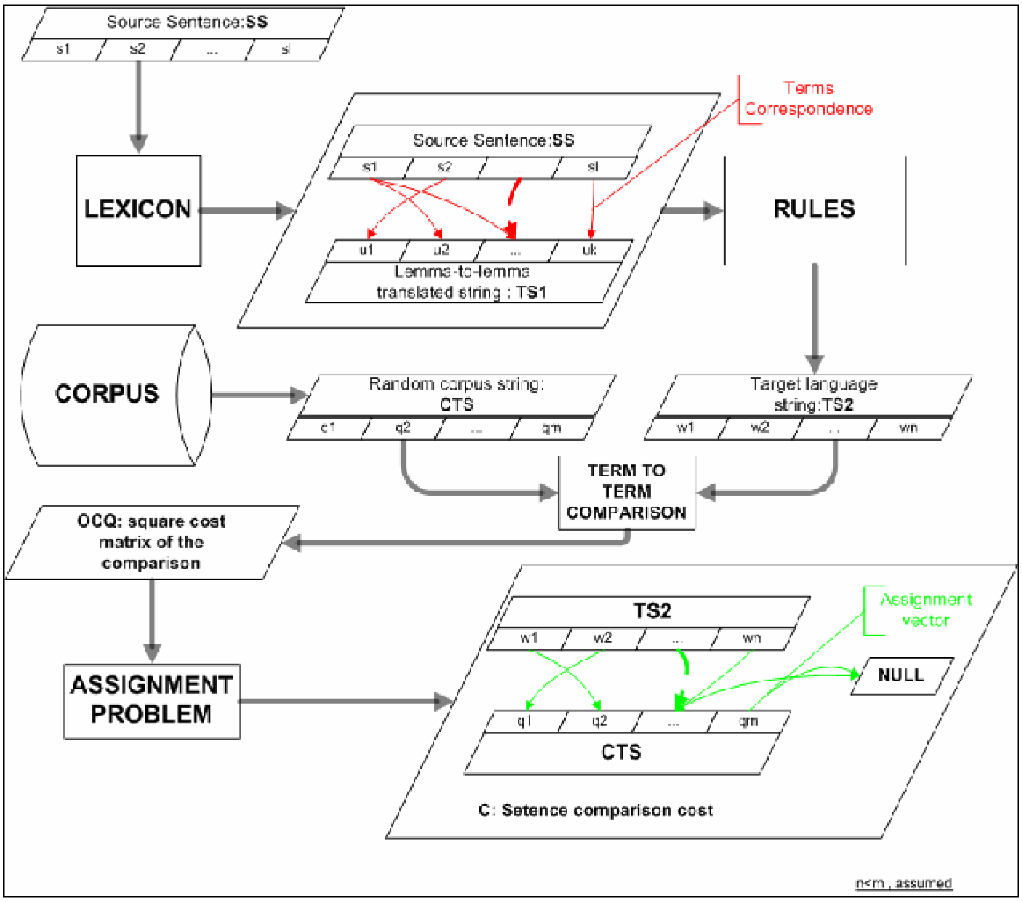

Figure 3 *- Complete diagram of the system*

In the next sections, the modules of the METIS system are described in detail.

3.1 Lemma-to-Lemma Translation

During this step the lemmata and the lexical phrases of the source language sentence (NS1) are translated to target language lemmata or lexical phrases by applying the bilingual lexicon. Each source language sentence is first scanned for matching and translating any lexical phrases occurring in it and then, the remaining lemmata are translated one by one. The result of the lemma-to-lemma translation is a target language sentence (TS1) where each of the source language lemmata corresponds to one or more lemmata in the target language sentence.

3.2 Rule Application

In a second step, METIS transforms the output string of the lemma-to-lemma translation process to a string whose structure is closer to that of the target language sentences sought. The fact that METIS does not use bitexts makes the structural transformation necessary for the matching techniques to return reasonable results.

Each METIS rule consists of a left hand part that encodes a source language pattern and a right hand part that describes the structural changes that will be imposed on TS1. An example rule (also mentioned in Section 2.2, Fig. 1) is presented in Fig. 4.

```
[1\ *\ AtDf*][\ *\ ^No*]{,2}[2\ *\ NoPr*] -> [2\ \ NP0];
```

Figure 4 *- Example rule*

The left hand part of the rule is occupied by a regular expression consisting of source language "term patterns" – each term pattern is surrounded by square brackets ([]) and consists of three fields. The first field of a term pattern is a number, here called "unification number", that is used to relate the match of the source language terms in NS1 with the transformations described in the right hand part (to be applied on the corresponding terms in TS1). The second and third field of the term patterns contain a list of positive and negative lemmata-patterns and a list of positive and negative tag patterns respectively. A negative pattern starts with a "^", while positive patterns have no prefix characters. In the example above the lemma field contains only the star character. In order to match a term-pattern, the values in the fields of an actual term must match the lemma and tag fields in the term pattern. The value of a field matches the corresponding value in a pattern list if it matches at least one of the positive entries and none of the negative ones.

The term-changers (that is, the elements of the right hand part of the rule) contain three fields: a number, a lemma list and a tag list. Once a term-pattern with unification number *a* has matched a term of NS1, the lemmata and the tags of the term-changers with the same unification number *a* will affect the corresponding target language terms in TS1. If a term-pattern with a unification number exists on the left hand side of the rule, while no term-changer with the same unification number exists on the right hand side of the rule, the terms in TS1 that correspond to the terms of the matching term-pattern are deleted. When a term-changer has no unification number, then a target language term is created and integrated in the TS1. When a term in NS1 corresponds to more than one term, for instance, when a lemma in the source language is translated into a phrase in the target language, a vector called "Main Tag Correspondence" or MTC is used which has the form shown in Fig. 5:

```
MTC:{Vb*,No*,At*}={V*,N*,AT*} ;
```

Figure 5 *- Example of MTC*

MTC indicates the corresponding term, which has to undergo the particular transformation. According to the MTC in Fig. 5, when a term NST1, containing a tag that matches Vb* in NS1, matches a term, the term-changer of the right hand part of rule affects only the corresponding terms of NST1 in TS1 that have a tag matching V*.

3.3 Corpus Search For Optimal Translation

After a target language sentence pattern with an improved structure has been created, the system tries to locate in the target language corpus the sentence whose structure is closer to the pattern provided.

In order to compare two sentences, METIS uses the metric that is described in the following section.

3.3.1 METIS metric for comparing sentences

Let **TS2** with terms *w1, w2, w3, w4,…,wn* be the result of the application of the mapping + transformation processes on a source language sentence **NS1** and **CTS** with terms *q1,q2,q3,q4,…,qm* be a random sentence in the target language corpus. Without loss of generality, we assume that $n<m$.[2] Every term *wi* of **TS2** is compared with every term *qk* of **CTS**. For each term pair, both the tags and the lemmata fields are compared. If any lemma in the *wi* term is identical with any lemma in the *qk* term, the lemma flag (or *lf*) is equal to 0. Compatibility of tags is expressed as the percentage of identical characters starting from the beginning and counting until the first non-identical character is found, divided by the length of the smaller of the two tags. The result of the tags-fields comparison is the highest percentage achieved from the compared pairs and its value ranges from 0 to 1. In table 4, examples of tag-field comparisons are shown:

***wi* tag-fields**	***qk* tag-fields**	**Result: *tf***
V-VHB	VVB	1
VVB	VHB	0.33
AJ-NN2	NN1	0.77

Table 4- *Tag-field comparison examples*

The overall cost of the comparison of the two terms is given by the following formula:

$$C= CR(1)+(lf*lcf +(1-ts)*tcf)*ocf*CL \quad (1)$$

Where

C: is the resulting cost of the comparison of the two terms

CR: is the cost range vector e.g [-200 200]

CL: is the length of the cost range vector CL=CR(2)-CR(1)

lf: Lemma equality flag

ts: Tag equality percentage

tcf: tag cost factor

lcf: lemma cost factor lcf=1-tcf

ocf: overall cost factor

The *ocf* parameter is used to weight the significance of the term in the whole sentence comparison procedure while the *tcf* and *lcf* parameters are used to weight the significance of the lemmata and tag field comparison in the term distance calculation. The way that *tcf* affects the comparison cost of *wi* and *qk* is obvious from the formula: the significance of the tag-fields increases as *tcf* increases, while the significance of the lemmata-fields decreases. The significance of the term with respect to the overall sentence comparison increases as *ocf* increases. *Ocf* and *tcf* change according to the tag of *wi*. There is a series of entries in the weights file (Section 2) that connect the tag patterns with the *tcf* and *ocf* values. The values that are connected with the first tag-pattern of the entries that match the tag of *wi* is used during the

[2] The only difference in the process if m<n is that some of the terms of TS2 will not correspond to any of the terms in any terms of CTS instead of the opposite.

comparison. As already mentioned, *wi* may have multiple tags that can possibly match different tag patterns. In such a case, the system compares *wi* with *qk* using every set of values that is retrieved from the weights file, and assigns the comparison result to the smallest cost. The lower limit of the comparison cost is s returned with the actual value of the comparison. This value is given by the following formula:

$$bcv=ocf*(CL+CR(1)) \quad (2)$$

Where

ocf : is the *ocf* value that was finally used for this comparison

CL, CR: as mentioned in the previous figure

Let ***OC*** be the *mxn* matrix that contains the results of comparing all the terms in **TS1** and **CTS**. The system will pad this matrix with an *mx(m-n)* matrix whose elements are all equal to *p*. The *p* value is the PadCost parameter in the weights file. Let ***OCQ*** be the *mxm* square matrix that is the result of the padding. The next step is to solve the assignment problem on the matrix ***OCQ*** and find the best assignment of terms *w1,…,wn* to the terms *q1,…,qm*. Let this be the vector *as* with size m. The first n elements of vector *as* show the best assignment of the n **TS1** terms to the **CTS** terms, while the terms from *n+1* to *m* show which of the terms in **CTS** do not match any of the terms in **TS1**. The overall cost of the comparison is:

$$ovc = \sum_{i=1}^{n} OCQ[as(i),i] + (m-n)*p \quad \textbf{(3)}$$

Where:

ovc: is the overall cost of the comparison

OCQ: the square cost matrix of the comparison

as: the assignment vector of **TS1** to **CTS**

m: the length of **CTS**

n: the length of **TS1**

p: the pad cost as retrieved from the weights file

In order to find the similarity percentage of the two terms, the smallest cost that the sentence could have for this assignment is calculated. For that reason the *mxn* matrix ***BOC*** is formed that has the *bcv* values that are the result of formula 2 for every comparison. Let BCQ be the mxm matrix that is constructed by padding ***BOC*** in the same way as ***OC***. The smallest cost of the given assignment is:

$$ovcb = \sum_{i=1}^{n} BCQ[as(i),i] + (m-n)*p \quad \textbf{(4)}$$

where:

ovcb: is the overall cost of the comparison

BCQ: the square smallest cost matrix of the comparison

as: the assignment vector of **TS1** to **CTS**

The similarity percentage is

$$sp = \frac{ovc - ovcb}{ovcb} * 100 \quad \textbf{(5)}$$

The system will return sentences of the corpus ordered by their similarity percentages.

3.4 Example

We will now show the process of the example source language sentence SS in table 5:

Word(Token)	**Lemma**	**Tag**
Η	ο	AtDfFeSgAc
γυναίκα	γυναίκα	NoCmFeSgNm
καθαρίζει	καθαρίζω	VbMnIdPr03SgXxIpAvXx
το	ο	AtDfNeSgAc
μήλο	μήλο	NoCmNeSgAc
.	.	PTERM

Table 5 *– Source Sentence (SS)*

The system will now search in the bilingual lexicon to find the translation of the source language lemmata. The lemma-to-lemma translation process results to sentence TS1 in table 6:

Lemma	**Tag**	**Corresponding SS Term**
the	AT?	1
woman	N??	2
peel	V??	3
the	AT?	4
apple	N??	5
.	PUN	6

Table 6 *– Lemma-to-lemma translated sentence (TS1)*

Tags marked on table 6 come from the lexicon (See Section 2.1). The lemma-to-lemma translation process also connects each of the terms of SS with one or more terms of TS1. The connection here is trivial because each of the lemmata of SS is translated in only term.

The next step of the translation process is the application of rules. Lets assume that the rules file is the following:

```
MTC:{Vb*}={V*};
[1\ \ VbMnIdPr______IpAv__] -> [1\ \ VVB] ;
```

Figure 6 *–Example Rules file*

The translation of present in Greek is present and present continuous. To make the example easier the rule on fig.6 describes only simple present. The term pattern on the LFH part of the rule will match the term [καθαρίζει\ καθαρίζω\ VbMnIdPr03SgXxIpAvXx] that is connected to the term [\ peel\ V??] of TS1.The term changer on the RGH part of the rule will change the term to [\ peel\ VVB]. On table 7 the result of the rule application is shown:

Lemma	**Tag**
the	AT?
woman	N??
peel	VVB
the	AT?
apple	N??
.	PUN

Table 7*-Sentence after rules application (TS2)*

We will now show how TS2 is compared to the target language sentence in table 8:

Word(Token)	**Lemma**	**Tag**
The	the	AT0
woman	woman	NN1
eats	eat	VVB
the	the	AT0
green	green	AJ0
apple	apple	NN1
.	.	PUN

Table 8 *–Example corpus sentence (CTS)*

The next step is to compare all terms in pairs to construct the **OC** matrix. To demonstrate this process we will show how the terms [eats\ eat\ VVB] and [\ peel\ VVB] are compared. The lf flag will be 0 because the two terms have different lemmata, while the tf will equal to 1 because the tags are identical. If the parameter values that are derived from the weights file for this comparison are ocf=1 and tcf=0.3 while the general variable **CL**=[-200 200] then the cost given by formula (1) is C=64. After we compare all terms of TS2 with the terms of CTS and we pad using the value p=60 (derived from weights file) then the result is the **OCQ** square matrix shown on table 9.

	the	**woman**	**peel**	**the**	**apple**	**.**	**<pad>**
the	**-20**	200	200	-20	200	200	60
woman	200	**-160**	200	200	0	200	60
eats	200	200	**64**	200	200	200	60
the	-20	200	200	**-20**	200	200	60
green	10	200	200	10	200	200	60
apple	200	0	200	200	**-160**	200	60
.	200	200	200	200	200	**-20**	60

Table 9 *– OCQ matrix and the solution of the corresponding assignment problem.*

The overall cost for this comparison given by formula 3 is ovc=-256.

5. Weights fine-tuning

The weights file contains a number of fixed parameters and an arbitrary number of *ocf* and *tcf* values that are connected with certain tag-patterns. The selection of the values of those parameters affects the selection and the order of sentences (the sentences are selected from the target corpus). In order to fine tune those parameters a large number of experiments is needed.

The results of each experiment have to be evaluated and the reasons behind the perceived behaviour of the system must be studied and explained. After that, any faulty values must be changed in order to obtain a better output. We have devised an evaluation method in order to compare the output sequence of METIS (retrieved from a limited and well-controlled corpus) with the sequence of best translations that was defined by a team of translators. The corpus was limited for this process because we wanted to contain only very similar sentences.

The experiments we conducted relied on corpora with very similar sentences that only differed in minimal details, for example in the tense of the verb. This enabled us to study the ability of the system to select among very similar target language sentences; clearly, it was easy for METIS to find the correct sentence in a set of irrelevant strings. In these experimental conditions, we have been able to fine-tune the system up to the point of receiving sequences with zero aberration from the sequence of the human translators.

An a example of the influence of parameters in the METIS system is given in fig. 6, that shows the change of the METIS performance is affected on the change of *tcf*.

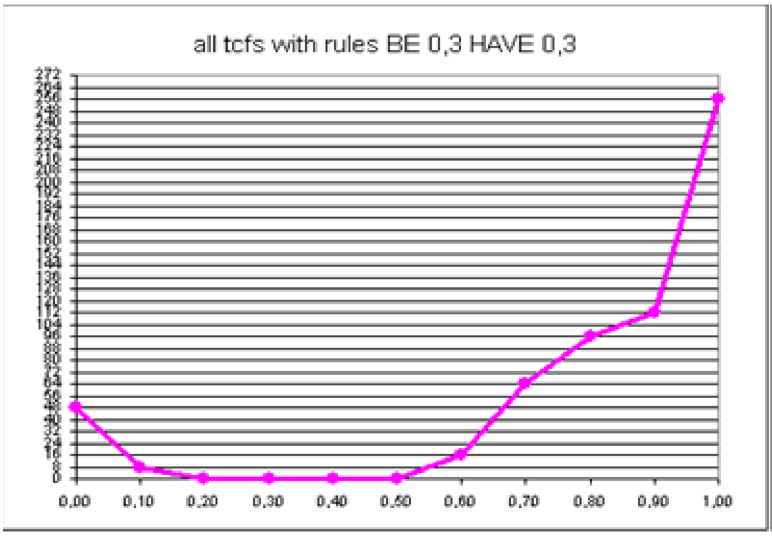

***Figure 6** – Metis performance vs. the tcf values of the weights file*

6. Future work

The main problem with METIS (and any MT system that retrieves translations from a corpus) is that a huge corpus is needed to ensure that a close enough match will be found even for the smallest sentences. One promising approach seems to be first to divide the sentence in chunks and then search for the chunk translations in the corpus. The chunk translations have to be put together in order to obtain the translation of the original sentence. Two problems present themselves with the chunk approach: (i) which is the best chunking? Probably, the best chunking will retrieve grammatically correct target language chunks from the corpus. Using some family of grammars is considered for this purpose (ii) how will the retrieved target language chunks be put together in order to form the translation of the original sentences? May be, another family of grammars is the answer to this problem.

7. Acknowledgement

This work is partially supported by the Future and Emerging Technologies (FET) unit of the IST program of the European Commission. The author is solely responsible for the content of this communication. It does not represent the opinion of the European Commission, and the European Commission is not responsible for any use that might be made of data appearing therein. The development and evaluation of the METIS system is the work of the METIS team with the following members: Stella Markantonatou, George Tambouratzis, Athanasia Fourla, Olga Yanoutsou and the author . The system is based on an original idea by Yannis Dologlou of the ILSP.

References

BNC: http://www.hcu.ox.ac.uk/BNC/

ILSP corpus: http://www.ilsp.gr/hnc_gr.html

Al-Onaizan, Yaser, Jan Curin, Michael Jahr, Kevin Knight,John Lafferty, Dan Melamed,Franz-Josef Och, David Purdy,Noah A. Smith and David Yarowsky, 1999. Statistical Machine Translation, Final Report, John Hopkins University.

Al-Onaizan., Yaser, Ulrich Germann, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Daniel Marcu and Kenji Yamada. 2000. "Translating with Scarce Resources". In *Proceedings of the 17th National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence*, July 30-August 3, Austin, Texas, pp. 672-678.

Brown, Peter F., John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Fredrick Jelinek, John D. Lafferty, Robert L. Mercer and Paul S. Roosin. 1990. "A Statistical Approach to Machine Translation". *Computational Linguistics*, Vol. 16, No. 2, pp 79-85.

Brown, Peter F., Stephen A. Della Pietra, Vincent J. Della Pietra and Robert L. Mercer. 1993. "The Mathematics of Statistical Machine Translation: Parameter Estimation". *Computational :Linguistics*, Vol. 19, No. 2, pp 263-312.

Dagan, Ido and Alon Itai. 1994. "Word Sense Disambiguation Using a Second Language Monolingual Corpus". *Computational Linguistics*, Vol. 20, No. 4, pp 563-596

Dologlou, Yiannis, Stella Markantonatou, George Tambouratzis, Olga Yiannoutsou, Athanasia Fourla and Nikos Ioannou, N. 2003. "Using Monolingual Corpora for Statistical Machine Translation: The METIS System". In *Proceedings of the EAMT- CLAW'03 Workshop*, Dublin, Ireland, 15-17 May, pp. 61-68.

Dorr, Bonnie J., Pamella W. Jordan and John W. Benoit 1999. “A Survey of Current Paradigms in Machine Translation”. *Advances in Computers*, Vol. 49, pp. 1-68. London: Academic Press

Dyvik, Helge. 1995. “Exploiting Structural Similarities in Machine Translation”. *Computers and the Humanities*, Vol. 28, pp. 225-234

Fourla, Athanasia, Olga Yannoutsou, Ioanna Tsakou , Sofia Stamou and Angeliki Petrits , 2000. “The contribution of a user group to the evaluation and improvement of an MT system”. *Translating and the Computer*, Vol. 22. London: ASLIB

Germann , Ulrich, Michael Jahr, Kevin Knight, Daniel Marcu and Kenji Yamada. 2001. “Fast Decoding and Optimal Decoding for Machine Translation”. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL 2001),* Toulouse, France, pp. 228-235.

Grefenstette, Gregory. (1995) “Comparing two Language Identification Schemes”. In *Proceedings of the 3rd International Conference on the Statistical Analysis of Textual Data (JADT’95),* Rome, Italy, Dec. 1995. Available at http://www.rxrc.xerox/com/publis/mltt/ jadt/jadt.html.

Johnson, Rod, Margaret King and Luis des Tombe. 1985. “EUROTRA: a Multilingual System under Development”. *Computational Linguistics*, Vol. 11, April-September 1985, pp. 155-169

Labropoulou, Penny, Elena Mantzari, and Maria Gavrilidou. 1996. Lexicon-Morphosyntactic Specifications: Language Specific Instantiation (Greek), PP-PAROLE, MLAP 63-386 report

Melamed, Dan I. 2001. Empirical Methods for Exploiting Parallel Texts. The MIT Press

Nagao, Makoto. 1984. “A Framework of a Mechanical Translation between Japanese and English by Analogy Principle”. In Alick Elithorn and Ranan Banerji (eds) *Artificial and Human Intelligence*. North-Holland

Papageorgiou, Harris , Prokopis Prokopidis, Voula Giouli, and Stelios Piperidis. 2000. “A Unified PoS Tagging Architecture and its Application to Greek”. In *Proceedings of The Second International Conference on Language Resources and Evaluation*, Athens, Greece, 31 May-2 June, pp. 1455-1462

Pentheroudakis, Joseph and Lucy Vanderwende. 1993. Automatically Identifying Morphological Relations in Machine-Readable Dictionaries. Technical Report MSR-TR-93-06, Microsoft Research Advanced Technology Division, Microsoft Corporation, One Microsoft Way, Redmond, WA. 98052.

Scott, Bernard E. 1989. “The LOGOS System”. In *Proceedings of MT SUMMIT II, pp.* 174- Alan Van den Bosch and Walter Daelemans. 1999. Memory-based morphological analysis. Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics, ACL'99, University of Maryland, USA, June 20-26, 1999, pp. 285-292.