A Formalized Reference Grammar for UNL-based Machine Translation between English and Arabic

Sameh Alansary [1,2]
(1) Bibliotheca Alexandrina, Alexandria, Egypt
(2) Faculty of Arts, Department of Phonetics and Linguistics, Alexandria University
El Shatby, Alexandria, Egypt
Sameh.alansary@bibalex.org

ABSTRACT

The Universal Networking Language (UNL) is an artificial language that can replicate human language functions in cyberspace in terms of hyper semantic networks. This paper aims to: a) design a reference grammar capable of dealing with the basic linguistic structures in order to act as a test-bed in automating translation between English and Arabic in both directions through UNL; b) evaluate the current state of the UNL system as an Interlingua in analyzing and generating English and Arabic as far as the reference structures are concerned. A reference parallel corpus of 500 structures was used. Results are promising; precision and recall of analyzing English to UNL (UNLization) are 0.979 and 0.96 respectively, while precision and recall of analyzing Arabic to UNL are 0.98 and 0.96 respectively. Precision and recall of generating English from UNL (NLization) are 0.97 and 0.96 respectively, while precision and recall of generating Arabic from UNL are 0.989 and 0.96 respectively.

KEYWORDS: Reference Grammar, Formal Grammar, Interlingua, UNL, UNL-ization Grammar, NL-ization Grammar, Machine Translation, Universal Networking Language, UNL system.

Introduction

While languages differ greatly in their “surface structures”, they all share a common “deep structure”; hence came the idea of creating a universal representation capable of conveying this deep structure while enjoying the regularity and predictability natural languages lack. Although interlingua is a promising idea, the number of interlinguas created is still very limited. Examples of well-known interlinguas are DLT (Witkam 2006), UNITRAN (Dorr (1987, 1990) and (Dorr et al. (2004)), KANT (Nyberg and Mitamura (1992), Nyberg et al. (1997)) and UNL (Uchida 1996, Uchida and Zhu (1993, 2005), Alansary et al. (2010)). The first three of these interlinguas lack standardization, however, the fourth, UNL, has succeeded in standardizing its tools, tagset and methodology as well as rely on meaning as an intermediate representation (Alansary 2011). UNL is a kind of mark-up language which represents the core information of a text. The UNDL Foundation, the founder of UNL, has created a wrapper application for development of various UNL tools and applications (Martins 2012, Martins and Avatesyan 2009). All engines, resources and tools are available through the UNLweb (www.unlweb.net) that contains many tools designed for linguists, computational linguists as well as non-professionals. These tools are used in analysing and generating natural languages. IAN, the Interactive ANalyzer, it employs the analysis grammar rules to analyze input and finally generate its corresponding UNL expressions. It operates semi-automatically; word-sense disambiguation is still carried out by a language specialist, nevertheless, the system can filter the candidates using an optional set of disambiguation rules. EUGENE (the dEep-to-sUrface natural language GENErator) is a fully automatic engine, it simply uses the target language grammar rules in order to decode the incoming UNL document and generate it in natural languages. IAN and Eugene use two types of Natural language dictionaries; enumerative and generative. The enumerative dictionary of IAN contains all inflected word forms of a language together with their corresponding Universal Words (concepts) and a set of linguistic features covering different linguistic levels. The generative dictionary, on the other hand, is the same as the ‘enumerative’ one but it contains all lexemes of language as bases together with a morphological paradigm number that controls the generative morphological behaviour (e.g. agreement and inflected forms) of words in natural language (Martins and Avetisyan 2009). It might be a fact that all languages have classical reference grammars in grammar books. Such a reference grammar maybe defined as a description of the grammar of a language, with explanations of the principles governing the construction of words, phrases, clauses, and sentences. It is designed to give someone a reference tool for looking up specific details of the language. In Natural Language Processing, computers should also learn a language in order to give a comprehensive and objective test-bed that enables us to evaluate, compare and follow up the performance of different grammars. A formalized reference grammar is needed in order to synchronize different languages; the UNL is initiating this idea as it utilizes a standardized environment. The current paper is limited to English and Arabic only; it is organized as follows: Section 1 discusses the design and compilation of the reference corpus. Section 2 discusses the design and implementation of the analysis grammar. Section 3 discusses the design and implementation of the generation grammar. Section 4 evaluates the analysis and generation results in English and Arabic. And finally section 5 is a conclusion and future work.

1 Reference corpus

Corpora are considered essential language resources necessary when building grammars. A reference corpus has been compiled as an experimental English corpus in order to prepare the initial version of analysis and generation grammars. An Arabic parallel Corpus has been compiled by translating the English reference corpus into Arabic, this corpus consists of 500 sentences collected from English grammar books. It is supposed to cover the basic and common linguistic phenomena between all languages that may be encountered in the process of building grammars within the UNL framework such as: temporary entries (e.g. URLs, nonsense words, symbols etc.), words that are not found in the dictionary (a grammar in NLP may face a set of words that might not be found in the dictionary), numerals, determiners, prepositions, conjunctions, noun phrase structures, expressions of time, verb forms, pronouns and sentence structures. The English reference corpus is manually annotated to make a standard version of UNL reference corpus. Both versions; English reference and UNL corpora, are available on the UNL web (http://www.unlweb.net/wiki/Corpus500).The Arabic UNL language centre has translated the English reference corpus into Arabic.

2 Building the UNLization (analysis) Grammar

UNLization is the process of representing the content of a natural language structure using UNL. In order to UNLize any Natural language text, the UNLization (analysis) grammar for that natural language should be, first, developed. The UNLization reference grammars for English and Arabic reference corpora have been already built to represent the content of both corpora. English and Arabic grammars have common modules such as; the tokenization, numeral, attribute, syntactic and syntax-semantic modules; however, the Arabic analysis grammar has an extra module; namely, the transliteration module which was developed in order to transliterate words that are not found in the Arabic Analysis dictionary into Latin characters. The following sub-sections will describe each of the aforementioned modules.

2.1 The Tokenization module

The tokenization algorithm is strictly dictionary-based; the system tries to match the strings of the natural language input against the entries existing in the dictionary. In case it does not succeed, the string is considered a temporary entry. There are no predefined tokens: spaces and punctuation marks have to be inserted in the dictionary in order to be treated as non-temporary entries. The tokenization algorithm goes from left to right trying to match the longest possible string with dictionary entries, and it assigns the feature **TEMP** (temporary) to strings that are not found in the dictionary. For instance, any URL such as "www.undlfoundation.org" should be considered TEMP; however, it is tokenized according to the entries found in the dictionary as [www] [.] [u] [nd] [l] [foundation] [.] [or] [g], which is incorrect since we expect the whole string to be treated as a single temporary entry. In order to avoid that, a disambiguation rule applies to consider any string a single node if followed by blank space or a full stop (is at the end of the sentence). The tokenization algorithm blocks the segmentation of tokens or sequences of tokens prohibited by disambiguation rules. Disambiguation rules are not only responsible for the segmentation of the input, but also responsible for choosing the word senses most appropriate to the context. For instance, “you” have two different realizations in the dictionary; the singular second person pronoun and the plural second person pronoun. In the sentence “you love yourself”, disambiguation rules should prevent the choice of the plural pronoun, thus, causing the engine to choose the singular pronoun if the verb is followed by a singular personal possessive pronoun.

2.2 The Numerals Module

Numerals in UNL are temporary UWs and should be represented in UNL as Digits between quotes. There are two cases in Numerals; they may be present in the input as digits in which case the engine will consider them as TEMP automatically, or, they may be written in letters, in the latter case the numerals module is activated. In order to handle numerals in both English and Arabic, both a dictionary and analysis rules are required. The numerals module is part of both the English and Arabic grammars. There are 4 types of numerals to be covered in this module: cardinal, ordinal, partitive and multiplicative. We will examine cardinal numbers first as they constitute the base for other types of numerals. There are many subsets of cardinal numbers such as units, tens, hundreds, thousands, millions…etc. The first step towards analyzing them is compiling a small dictionary that will enable rules to convert numbers in both English and Arabic into digits. Some cardinal numbers will be inserted in the dictionary as is such as the numbers from one to nineteen. Other numbers will be inserted incomplete in the dictionary to be later completed by rules; for example, tens are inserted without their zeros, "twenty" is inserted as "2"….etc. The second step is to develop the required rules; units and numbers from ten to nineteen are retrieved from the dictionary without any modification by rules. Tens starting from the number twenty have two possibilities in analysis: the first is adding tens to units; for instance in the case of "twenty one", "twenty" which is stored in the dictionary as "2' and "one" which is stored as "1" will be joined by a rule and will be treated as a single number "21". The second is not adding tens to units as in "twenty", a zero will be added to "2" and joined together by a rule to become "20". The analysis of partitive numerals depends on their existence in the dictionary. In ordinal and multiplicative numbers; after converting the numbers in letters into digital numbers, an attribute "@ordinal" will be assigned to the number. If the number is followed by the word "times" such as "four times", the attribute "@times" will be assigned to "4"to be "4.@times".

2.3 Attributes module

In UNL, attributes have been used to represent information conveyed by natural language grammatical categories (such as tense, mood, aspect, number, etc). The set of attributes, which is claimed to be universal, is defined in the UNL Specs (http://www.unlweb.net/wiki/Attributes). The attributes module can handle determiners, pronouns, prepositions and verb forms. It is responsible for substituting certain words or morphemes with attributes, as in the case of quantity quantifiers ("a lot of", "several", "few", "all", "any…etc.) which will be deleted and substituted by the attributes "@multal, @paucal, @any, @all ..etc." to be assigned to the following word. In UNL, pronouns are "empty concepts" represented semantically as "00". The person, number and gender of the pronoun are described by UNL attributes.

2.4 Syntactic module

After assigning the necessary attributes, the syntactic module should start drawing the syntactic trees for noun phrases, verb phrases and sentence structures that are part of the corpus, according to the X- bar theory (http://www.unlweb.net/wiki/X-bar_theory). The syntactic modules for Arabic and English grammars both follow the same methodology, thus, the following subsections will present and discuss only English examples since they will be easier to understand. The syntactic module is divided into two phases; the list–to-tree phase and the tree-to-tree phase.

2.4.1 The list-to-tree phase

In this phase, rules are used to parse the tokenized input sentences into a tree structure. It starts by composing small trees for the small phrases in the sentence and then combining these small trees together to form a bigger tree. List-to-tree rules are responsible for building the trees for language structures; ordering of rules is required; rules for building noun phrase trees should be followed by rules for building verb phrase trees. The sentence "he buys books about cars from Paris" after being processed by the attribute module will be "he.@3.@male buy.@present book.@pl about car.@pl from Paris". This sentence has two noun phrases, rules will start to project all nouns and pronouns in the sentence to the intermediate constituent "NB" and link the NBs to form a higher constituent from right to left. "cars" and "Paris" will be firstly linked together under the intermediate constituent "NB" (A decision to link "from Paris" to the nearest head " car " was taken to solve the structural ambiguity concerning the PP attachment " from Paris ".) and projected to the maximal projection NP by linking it to an empty node which will act as the specifier. The preposition "from" between the two NBs will be deleted leaving the semantic feature "frm" to be assigned to "Paris". This feature will help later in the process of linking them with a semantic relation as will be shown in sub-section 2.6. then the NB "book" and the NP "cars from Paris" will be linked together under the bigger intermediate constituent "NB" and the preposition "about" will be deleted leaving the semantic feature "cnt" assigned to the NP "cars from Paris". Then, rules will project the NB "book about cars from Paris" to the maximal projection NP by linking it with an empty node which will act as the specifier, and the NB of the pronoun "he" will also be projected to an NP since a pronoun does not need a specifer, as shown in figure 1.

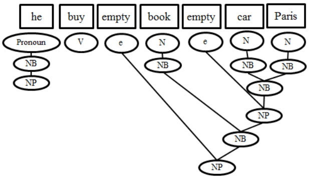


FIGURE1 – The syntactic tree for "books about cars from Paris".

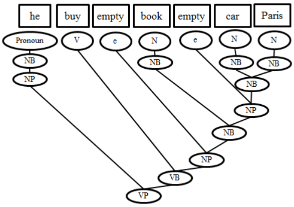


FIGURE2 – The syntactic tree for "he buys books about cars from Paris".

After building the tree of the noun phrase "books about cars from Paris", rules will start to form a bigger tree that contains this noun phrase and the verb "buy" by combining them under the intermediate constituent "VB". The transitive verb "buy" is carrying the feature "TSTD" in the dictionary; this means that it needs two syntactic arguments: a subject which is the doer of the action, and a single direct object. These two arguments are mapped semantically to an agent and an object. The two arguments should be "NPs", thus, the semantic feature "obj" will be assigned to the NP "books about cars from Paris". The biggest tree will be formed by combining the "VB" with the NP "he" under the maximal projection "VP" as in figure 2 and the semantic feature "agt" will be assigned to the pronoun "he".

2.4.2 The tree-to-tree phase

The constituents "NB, NB, NP, VB and VP" will be mapped onto their syntactic roles. The methodology used in this phase is to start decomposing constituents from the biggest to the smallest. Ordering of rules is required; rules for decomposing verb phrase trees should be followed by rules for decomposing noun phrase trees. VP is the last constituent to be composed, hence, it will be the first one to be decomposed. A key assumption of X-bar theory is that branching is always binary, thus, the decomposition of any constituent will affect the tree. A constituent is decomposed into a syntactic role between a node and the head of the adjacent constituent to make the binary relation. In the present example, the decomposition of "VP" will affect the tree; a "VS" or a Verb Specifier relation will be constructed between the verb and the pronoun as shown in figure 3. The second branch to be decomposed is the "VB" which will be decomposed into a Verb Complement relation "VC" constructed between the verb and the head of the NP "book about cars from Paris", "book", as shown in figure 3. After first decomposing the biggest constituents in the tree, the VP and the VB, decomposing the smaller NPs and NBs starts. Because the specifier slot in the noun phrase is empty, the NP syntactic relations between the empty nodes and the NBs will be deleted. The bigger NB "book about cars from Paris" will be decomposed into "book" and the head "cars" of the smaller NB "cars from Paris", and the syntactic relation "NA" or Noun Adjunct will be established between them as shown in figure 4. Finally, the smallest constituent in the tree will be decomposed into the two nouns "car" and "Paris", and the syntactic relation "NA" will be constructed between them as shown in figure 4. The output of the tree-to-tree phase; the four syntactic relations: VS, VC, NA and NA will be the input of the tree to Network phase.

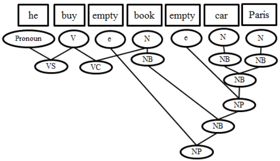


FIGURE3 – Constructing the "VS" and "VC"

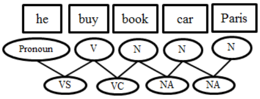


FIGURE4 – Constructing the "NAs"

2.5 Syntax – semantic module

In this module, rules have been built to derive the semantic network from the syntactic graph. Order of rules in this module are not necessary since the semantic features assigned to nodes from the list-to-tree phase (see section 2.4.1) constrain the rules enough to be carried out in their context. In the present example, the output of the tree-to-tree phase will be the input of this module. In this module, the VC, VS, NA and NA will be mapped with their corresponding semantic relations: "obj", "agt", "cnt" and "frm" respectively .

3 Building the NLization (Generation) Grammar

This section discusses the NL-ization of the reference corpus from the interlingua representation (UNL) into both Arabic and English. To achieve this purpose, Arabic and English linguistic resources have been developed. These resources are Arabic and English specialized dictionaries in addition to Arabic and English NL-ization grammars. The process of generation may be seen to some extent as a mirror image of the analysis process; generating well-formed sentences has to pass through a set of grammar modules which are: the semantic-syntactic module, the syntactic module, the attributes module, the numerals module and also a transliteration module that is responsible for to transliterating temporary UWs that are not found in the Arabic generative dictionary into Arabic characters.

3.1 The Semantic-Syntactic Module (Network-to-Tree Phase)

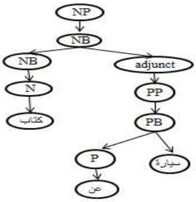
This module is responsible for mapping the semantic relations onto their syntactic equivalents. As an example, the semantic graph generated in section 2.5 representing a verbal phrase requires mapping rules to map the semantic relations agt, obj, cnt, and frm onto their counterpart syntactic relations; Verb specifier (VS), Verb Complement (VC), Noun Adjunct (NA) and another Noun Adjunct (NA) respectively. Moreover, in case the semantic relations "cnt" and "frm" are the counterpart of the syntactic relation noun adjunct (NA), mapping rules should also take into consideration whether the noun adjunct requires a preposition or not. The generated syntactic relations will be processed in the following section 3.2.

3.2 The Syntactic Module

The syntactic module is the second module of the NL-ization grammar, it is responsible for transforming the deep syntactic structure generated from the semantic-syntactic module into a surface syntactic structure. The Syntactic module is divided into two phases; the tree-to-tree phase and the tree-to-list phase. The tree-to-tree phase is responsible for gathering individual syntactic relations and forming higher constituents while the tree-to-list phase is responsible for linearizing the surface tree structure into a list structure. The following two subsections will explain these two phases in more detail.

3.2.1 The tree-to-tree phase

In the tree-to-tree phase, rules are responsible for building the surface syntactic structure of the sentence by building the intermediate constituents (XBs) which are combined to form the maximal projections (XPs) and finally combined to form the sentence structure. For example, the syntactic relations VS, VC, and the two NAs will be combined to form the maximal projection VP according to the schema of X-bar theory. The NA between "كتاب" and "سيارة" will be transformed gradually to the maximal projection NP passing through the intermediate projection NB as shown in figure 5, the second NA between "سيارة" and "باريس" will also become a NP as shown in figure 6. In figure 5, the preposition (P) "عن" was inserted in the tree as the adjunct of the noun "كتاب", "كتاب"in the current example needs a preposition which is predicted by means of the semantic – syntactic module. Similarly, the preposition "من" was inserted in figure 6. The NP in figure 5 is combined with the NP in figure 6 to constitute the complement of the main verb "اشترى" as shown in figure 7.



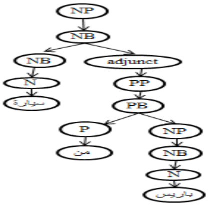


FIGURE 5 – The maximal projection for NA

FIGURE 6 – The maximal projection for NA

The verb complement will in turn be combined with the verb "اشترى" to form the intermediate projection VB "اشترى كتاب عن سيارة من باريس". Finally, the resulting VB is combined with the specifier (VS) to build the final maximal projection of the phrase VP as shown in figure 8.

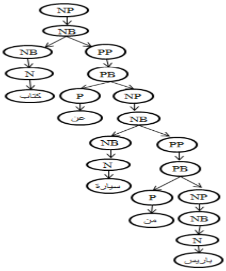


FIGURE 7 – combined the two NPs

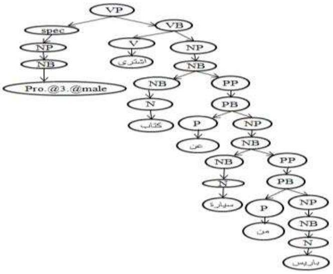


FIGURE 8 – the syntactic tree of the final VP

3.2.2 The tree-to-list phase

In the tree-to-list phase, rules are responsible for transferring the surface syntactic structure into a list structure and also adding the required spaces. Thus, the syntactic tree in figure 8 will be transformed into the natural language list "هو اشترى كتب عن السيارات من باريس". "كتب" and "سيارات" were generated as plural forms since they carried the attribute "@pl" in the semantic network. Attribute assigning will be discussed in subsection 3.3.

3.3 The Attributes Module

This module is responsible for converting the attributes represented in the interlingua into the suitable natural language words or affixes. For example, pronouns are represented in UNL as "00" nodes along with some attributes to reflect their number, gender....etc. The pronoun in figure 9 will be replaced by "**هو**"[1] in the list structure as shown in section 3.2.2. Moreover, there are many types of attributes represented in the UNL framework; all are handled during the NLization process. For example, an attribute expressing definiteness such as "@def" will be realized as the prefix "ال", and an attribute expressing number such as "@pl" may be realized as a suffix such as "ات".

3.4 The Numeral Module

This module is responsible for converting digital numbers onto their counterpart natural language string (Arabic or English). There are four types of numerals to be covered in the numerals module; cardinal numbers, ordinal numbers, partitive numbers and multiplicatives. For cardinal numbers, the basic conversion mechanism is converting individual digits from (0 to 9) directly onto the counterpart natural language string, and then converting multiple digits by combining the converted individual digits to form bigger numbers. The numerals module handles also ordinals, multiplicatives and partitive numbers.

4. Evaluation

The output of the UNLization process for both Arabic and English languages has been evaluated based on a corpus that is annotated manually semantically in order to figure out the quality and

[1] A decision was taken to generate an overt pronoun to make the structure more explicit.

accuracy of the automatically generated semantic networks. The output of the NLization process has been evaluated based on a manually translated corpus. The F-measure (F1-score) is used to measure of the grammar accuracy, according to the formula: F-measure = 2 x ((precision x recall) / (precision + recall)). Precision measurement of the UNL-ized Arabic sentences was 0.98 while recall measurement was 0.96. Precision measurement of the UNL-ized English sentences was 0.979 while recall measurement was 0.96. Also, the same measurement was applied to figure out the correctness of the automatically generated Arabic and English languages from the UNL-ized documents; the precision measurement of the generated Arabic was 0.989 while recall measurement was 0.96. The precision measurement for the generated English was 0.97 while the recall measurement was 0.96. Accordingly, the F-measure of English-UNL is 0.969, Arabic-UNL is 0.974, UNL-English is 0.964 and UNL-Arabic is 0.974. The values report a very high similarity between the actual output and the expected output.

5 Conclusion and Future Work

This paper presented a formalized reference grammar for analyzing and generating Arabic and English within the UNL framework. The design of the reference grammar depended on linguistic phenomenon common to all languages in order to support the idea of an Interlingua. The evaluation of the current state reflected very high accuracy which can: first, be the base of a more robust system of machine translation; second, a support for other languages in the UNL system in order to synchronize themselves by building parallel corpora and analysis and generation grammars. This would also constitute objective criteria to compare results. UNL as an Interlingua is expected to be used in several different tasks such as text mining, multilingual document generation, summarization, text simplification, information retrieval and extraction, sentiment analysis etc. Future work will be mainly directed to the reference corpus. It is planned to increase the number of structures from 500 to 1000, 5 sentences at least for every structure. Therefore, the minimal number of sentences to be processed in the next stage is expected to be 5000.

References

Alansary, S. (2010). A Practical Application of the UNL+3 Program on the Arabic Language. In Proceedings of the 10th International Conference on Language Engineering, Cairo, Egypt.

Alansary, S. (2011). Interlingua-based Machine Translation Systems: UNL versus Other Interlinguas. In Proceedings of the 11th International Conference on Language Engineering, Cairo, Egypt.

Alansary, S., Nagi, M., Adly, N. (2011). Understanding Natural Language through the UNL Grammar Work-bench , Conference on Human Language Technology for Development (HLTD 2011), Bibliotheca Alexandrina, Alexandria, Egypt.

Alansary, S., Nagi, M., Adly, N. (2010). UNL+3: The Gateway to a Fully Operational UNL System , 10th International Conference on Language Engineering, Ain Shams University, Cairo, Egypt.

AlAnsary, S. (2011), Interlingua-based Machine Translation Systems: UNL versus Other

Interlinguas. 11th International Conference on Language Engineering , Ain Shams University, Cairo, Egypt.

Alansary, S. (2012). A UNL-based approach for building an Arabic computational lexicon, the 8th international conference on informatics and system (infos2012), Cairo, Egypt.

Boitet C. (2002). A rationale for using UNL as an interlingua and more in various domains. Proc. LREC-02 First International Workshop on UNL, other Interlinguas, and their Applications, Las Palmas, 26-31/5/2002, ELRA/ELDA, J. Cardeñosa ed., pp. 23—26.

Dorr, Bonnie J. (1987). UNITRAN: An Interlingua Approach to Machine Translation. Proceedings of the 6th Conference of the American Association of Artificial Intelligence, Seattle, Washington.

Dorr, Bonnie J. (1990). "A cross-linguistic approach to translation". Proceedings of 3rd International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Language, Linguistics Research Center, University of Texas, Texas.

Dorr, Bonnie J., Hovy, E., Levin, L. (2004). Machine Translation: Interlingual Methods, Encyclopedia of Language and Linguistics. 2nd ed., Brown, Keith (ed.).

Nyberg E.H., Mitamura T. (1992). The KANT System: Fast, Accurate, High-Quality Translation in Practical Domains, in Proceedings of the International Conference on Computation Linguistics, (COLING 1992), Nantes, France.

Nyberg E. H., Mitamura T., Carbonell J. (1997). The KANT Machine Translation System: From R&D to Initial Deployment, in Proceedings of LISA (The Library and Information Services in Astronomy) Workshop on Integrating Advanced Translation Technology, Hyatt Regency Crystal City, Washington D.C.

Martins, R., Avetisyan, V. (2009). Generative and Enumerative Lexicons in the UNL Framework, the seventh international conference on computer science and information technologies (CSIT 2009), 28 September – 2 October, 2009, Yerevan, Armenia.

Martins, R. (2012). Le Petit Prince in UNL, the 8th international conference on language resources and evaluation (LREC'12), 23-25 May 2012, Istanbul, Turkey.

Uchida .H , Zhu .M. (2005). UNL2005 for Providing Knowledge Infrastructure, in Proceedings of the Semantic Computing Workshop (SeC2005), Chiba, Japan, 2005

Uchida H., Zhu M., (1993). Interlingua for Multilingual Machine Translation, in Proceedings of the Machine Translation Summit IV, Kobe, Japan.

Uchida H. and M. Zhu (2005). UNL2005 for Providing Knowledge Infrastructure, in Proceedings of the Semantic Computing Workshop (SeC2005), Chiba, Japan.

Uchida H. (1996). UNL: Universal Networking Language – An Electronic Language for Communication, Understanding, and Collaboration, UNU/IAS/UNL Center, Tokyo, Japan.

Witkam T. (2006). History and Heritage of the DLT (Distributed Language Translation) project, Utrecht, The Netherlands: private publication.