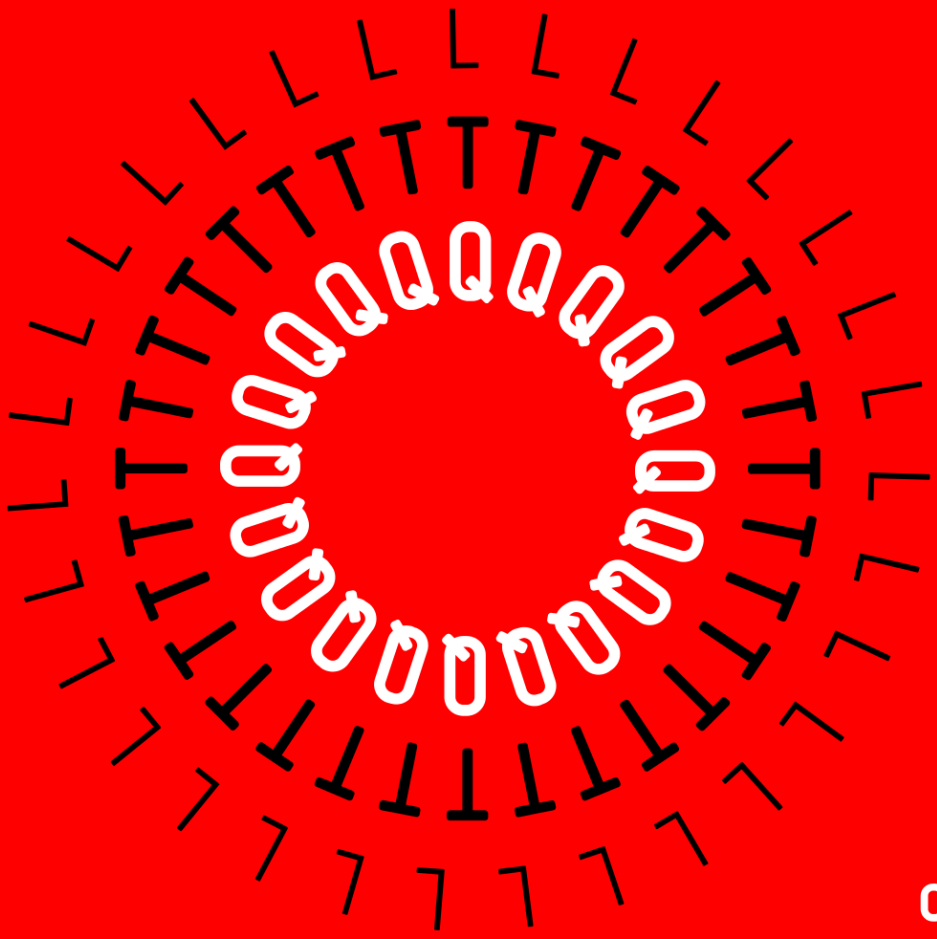


qtleap
quality translation by deep language engineering approaches

Report on the State of the Art Concerning Generation

DELIVERABLE D4.4

VERSION 5.0 | June 15, 2015

QTLeap

Machine translation is a computational procedure that seeks to provide the translation of utterances from one language into another language.

Research and development around this grand challenge is bringing this technology to a level of maturity that already supports useful practical solutions. It permits to get at least the gist of the utterances being translated, and even to get pretty good results for some language pairs in some focused discourse domains, helping to reduce costs and to improve productivity in international businesses.

There is nevertheless still a way to go for this technology to attain a level of maturity that permits the delivery of quality translation across the board.

The goal of the QTLeap project is to research on and deliver an articulated methodology for machine translation that explores deep language engineering approaches in view of breaking the way to translations of higher quality.

The deeper the processing of utterances the less language-specific differences remain between the representation of the meaning of a given utterance and the meaning representation of its translation. Further chances of success can thus be explored by machine translation systems that are based on deeper semantic engineering approaches.

Deep language processing has its stepping-stone in linguistically principled methods and generalizations. It has been evolving towards supporting realistic applications, namely by embedding more data based solutions, and by exploring new types of datasets recently developed, such as parallel DeepBanks.

This progress is further supported by recent advances in terms of lexical processing. These advances have been made possible by enhanced techniques for referential and conceptual ambiguity resolution, and supported also by new types of datasets recently developed as linked open data.

The project QTLeap explores novel ways for attaining machine translation of higher quality that are opened by a new generation of increasingly sophisticated semantic datasets and by recent advances in deep language processing.

www.qtleap.eu

Funded by
QTLeap is funded by the 7th Framework Programme of the European Commission.

Supported by
And supported by the participating institutions:

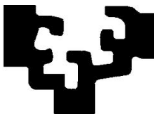

Faculty of Sciences, University of Lisbon


German Research Centre for Artificial Intelligence

Charles University in Prague

Bulgarian Academy of Sciences

Humboldt University of Berlin


University of Basque Country


University of Groningen

Higher Functions, Lda

Revision history

Version	Date	Authors	Organisation	Description
1.0	Feb 18, 2014	Dieke Oele and Gertjan van Noord	UG	First draft
2.0	Feb 24, 2014	Dieke Oele, Gertjan van Noord, Kiril Simov, Petya Osenova, Aljoscha Burchardt, Kepa Sarasola, Koldo Gojenola and Gorka Labaka	UG, IICT-BAS, DFKI, UPV/EHU	Feedback from IICT-BAS, DFKI and UPV/EHU integrated
3.0	Feb 27, 2014	Dieke Oele, Gertjan van Noord, Kiril Simov, Petya Osenova, Aljoscha Burchardt, Kepa Sarasola, Koldo Gojenola, Gorka Labaka, and Rosa Del Gaudio	UG, IICT-BAS, DFKI, UPV/EHU, HF	Feedback from HF integrated
4.0	Oct 02, 2014	Dieke Oele	UG	Added chapter 'Expected benefits for real user scenario'
5.0	Oct 17, 2014	Dieke Oele	UG	Feedback from partners integrated
6.0	June 2, 2015	Dieke Oele	UG	Feedback from commission integrated
7.0	June 10, 2015	Dieke Oele	UG	Feedback from partners integrated

Statement of originality
This deliverable contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Report on the State of the Art Concerning Generation

DOCUMENT QTLEAP-June-D4.4
EC FP7 PROJECT #610516

DELIVERABLE D4.4

completition
FINAL
status
RESUBMITTED
dissemination level
PUBLIC

responsible
Kiril Simov (WP4 Coordinator)
reviewer
Rosa Del Gaudio
contributing partners
UG, DFKI, IICT-BAS and UPV/EHU

authors
Dieke Oele, Gertjan van Noord, Kiril Simov, Petya Osenova, Aljoscha Burchardt, Kepa Sarasola, Koldo Gojenola, Gorka Labaka, and Rosa Del Gaudio

© all rights reserved by FCUL on behalf of QTLeap

Contents

1 Introduction

The task of Natural Language Generation (NLG) deals with the generation of target language texts. In this project a transfer system is used where a deep linguistic analysis of the source sentence is transformed into a deep linguistic representation serving as a basis for the construction of actual sentences. The generation procedure, in this context, requires not only the implementation of algorithms which construct sentences that are consistent with a given deep structure, but also the development of a statistical language modeling component to ensure that the most fluent candidate sentence is selected in such cases.

According to Reiter and Dale (2000), the typical architecture in NLG includes three major steps. In the first phase, the so-called macro-planning, it is determined what must be verbalized. In the context of machine translation, this part of the process usually is a result of processing the source text into a deep representation. The second phase, the so-called micro-planning, determines how the text should be verbalized. This step will ensure that the created semantic structure conforms to all the requirements for a complete semantic structure with respect to the grammar formalism. This procedure involves mapping the input to grammatical functions, including language dependent information, and recovering non-explicit information from the context (mainly document context, but could also be world knowledge context like Linked Open Data). Ultimately, in the realization step, actual sentences will be produced on the basis of the grammar. This is done by selecting appropriate lexical units and applying syntactic rules from the grammar. The goal of this component is to take the text specification, produced by the micro-planner, and convert it into text. The linguistic realization of abstract representations, such as abstract syntax, generally requires the use of a grammar, a formal description of the syntactic and morphological resources available in the output language.

2 Grammar-based models

Various Natural Language Generation systems have been proposed that differ in both complexity and sophistication. Simpler techniques, like systems that generate from canned text, only retrieve predefined stored text (Hovy et al. (1996)). Systems based on templates on the other hand, are more refined. Although they still make use of predefined stored text, they also include simple transformations. The problem of both systems is that they are neither flexible nor reusable. More advanced systems are based on features containing distinctions within the language. Grammar-based systems are a subcategory of these kinds of approaches. They employ linguistic constraints where the relations between meaning and form are encoded in an externally specified grammar. Here the task of the generator is to construct possible sentences based on a deep grammar and, subsequently, select the best sentence, for instance by use of a statistical language model.

2.1 Sentence construction

The first step in the process of generation is the construction of grammatical sentences from a given abstract representation. Various abstract sentence representations were proposed as input for the grammar-based models. In what follows, two of them will be discussed: Minimal Recursion Semantics (MRS) and Dependency Structures.

2.1.1 Minimal Recursion Semantics

One possible abstract representation that could be used as input for the generation of sentences in machine translation is Minimal Recursion Semantics (MRS) (Copestake et al. (1995), Copestake et al. (2006)). This is a flat, event-based, representation of semantics that nests semantic structures as a set of relations, without losing dependencies between these relations. In MRS, the core could be apprehended as a flat set of multiple elementary predications complemented by a 'handle' of the predication with the highest prominence and a set of 'handle constraints' that record restrictions on scope relations in terms of dominance relations. Because it is possible in MRS to generalize over classes of predicates and it allows for the under-specification of scope relations, it is considered an attractive input representation for the realization of sentences as linear text (Copestake et al. (2006)). Furthermore, it enables constraint-based semantic composition and can be implemented in typed feature structures. The LOGON machine translation system for Norwegian and English (Oepen et al. (2004), Velldal and Oepen (2005), Velldal and Oepen (2006a)) performs semantic transfer based on Minimal Recursion Semantics derived by an HPSG grammar. Following the transfer fase, the semantic representations are passed on to the generator that then produces sentences in the target language, operating from the semantic input representations. The target sentences are generated in a so-called Linguistic Knowledge Builder system (LKB, Copestake (2002)). This sentence realizer employs a lexically driven approach, which is suitable for grammars based on lexicons such as HPSG, where most of the information is encoded directly in lexical entries or lexical rules. The grammar used in LOGON is the LinGO English Resource Grammar, which is an implementation of HPSG with a fairly complete lexical and grammatical coverage over a variety of domains. The distribution includes treebanked versions of several reference corpora and provides disambiguated MRS-representations for each input sentence.

In LOGON a chart-based generator creates sentences from the MRS. The process of chart generation is very similar to chart parsing. In this task, however, the covering of edges is defined in terms of semantics, rather than orthography. As stated before, an MRS structure is primarily composed of a bag of elementary predications, which in turn define relations in MRS. For the realization of sentences in LOGON, a pre-processing phase first stores the lexical entries and the lexical grammar rules, based on their semantics. Then the chart generator initializes the chart by retrieving the lexical entries from the target language lexicon. If the lexical rules lead to relations, their application is only permitted if these relations correspond to parts of the input semantics. Next, lexical and morphological rules are applied to the lexical entries. The empty chart is then populated with edges that correspond to the instantiated entries and rules, each pointing to the semantic relations of the input that they cover. The second step is the actual generation of the charts. Inactive edges are matched against existing active edges or a novel active edge is created by matching an inactive one against the head daughter of a rule. Before two edges are combined in a construction, the generator first checks if the relations they cover do not overlap. Generation is complete when all inactive edges covering the entire input MRS have been found. The result of the chart generation process is a compact representation of all the possible realizations in the form of a forest structure.

2.1.2 Dependency structures

Another way to represent abstract linguistic input for the realization of sentences is Dependency Structure (Hays (1964)). A Dependency Structure represents a sentence as a

set of relations connecting all the words in a Dependency Tree. These representations are more abstract in comparison to syntactic trees since they do not constrain or prescribe a particular word order and have no explicit notion of constituents (Ambati (2008)). They are therefore more specific in terms of semantics and the notion of relations across words is more explicit. Another advantage of Dependency Structures is, similar to MRS structures, the fact that they are able to represent long distance dependencies between words.

One of the first attempts to implement dependency trees in machine translation is Lin (2004), who proposed a path-based transfer model, using a word-aligned parallel treebank. The training of the model results in a set of transfer rules that provide the corresponding translation fragment of the target language sentence, given a certain path in the source language. When all the words are linked with the target language words, rules are extracted that describe the dependency relations for the target language and their order based on the grammar.

Another approach to the implementation of dependency trees in machine translation is the 'treelet system' proposed by Quirk et al. (2005). A treelet is a sub tree of a dependency tree that is connected arbitrarily. After parsing the training data, the resulting dependency trees are projected onto the target language by use of word alignments. All words that break the linear sequence of the target sentence in the dependency tree with the lowest possible node are reattached. This way the order can be attained in relation to the siblings. Then the translation rules, or dependency treelet pairs, are extracted. Next, a decoder applies a bottom-up decoding strategy over the source dependency tree with the treelet pairs. A log-linear model that entails typical features such as language models, word alignment probabilities and reordering models scores the generated translation hypotheses.

De Kok et al. (2011) and De Kok (2013) exploit a combination of the use of HPSG and Dependency structures as input for a generation system. In their system, Dutch sentences are transformed into dependency structures using the Alpino parser (van Noord (2006)). This parser includes an attribute-value grammar inspired by HPSG, a large lexicon, and a maximum entropy disambiguation component. The realizer uses the same dependency structure as the Alpino parser, excluding information about word order and original word inflection. For generation, the grammar is used in the reverse direction as for parsing. Here generation process starts with creating possible sentences from the Dependency Structure. It comprises a syntactical representation of the sentence that needs to be realized in the form of a tree. Each node in the tree contains information about its main word and each dependent can by itself contain a dependency structure. When it is possible to have multiple dependents of the same type, Alpino uses a list structure. For the actual creation of sentences, a bottom-up chart generator is used. In order that the generator constructs partial analyses that are realizing the input dependency structure, top-down guidance is used. This setup requires that every category that is considered during the generation process includes a dependency structure unified with a part of the input dependency structure. During generation, incomplete realizations are packed in a realization forest for efficiency. Afterwards, full realizations can be obtained from the packed representation.

2.2 Sentence selection

The previously mentioned systems often generate several hundreds and thousands of different sentences for a given input. Previous work tried to find an effective way of ordering all these alternative hypotheses and, ultimately select the preferred final output string. This task is often referred to as realization ranking. In order to select the most fluent sentence from a set of candidate sentences, several statistical models have been examined in the past.

2.2.1 N-gram model

One of the first generation systems based on statistics is NITROGEN, created by Langkilde and Knight (1998). This hybrid system for Japanese-English machine translation employs N-gram models to rank symbolically over-generated lattices of possible output sentences. This method has been derived from work in speech recognition, where the realization task is split in two parts. First, a lot of hypotheses are created for the sentence to be generated by use of a basic grammar and, subsequently, the best of the sentences is chosen with a lattice search procedure. For the construction of the sentences, a small expansion grammar was built where each rule expands only a small part of the input resulting in several rules for the creation of phrases. Since the grammar is very broad, it severely overproduces sentences. For the input of the grammar, an abstract meaning representation is used, composed of concepts from the SENSUS Knowledge base, a classification of about 90,000 concepts derived from WordNet, and keywords relating these concepts to each other. The abstract meaning representation consists of a labeled directed graph, in which concepts can be linked with each other by nesting them to form more complex meanings. The associations between conceptual meanings are also marked through keywords in order to create the freedom to express the relations at various semantic and syntactic levels. In the generation process, the grammar is matched against the representation to find all related English words, which are then connected with the surface-level phrases to make sentences. This method ensures that even a simple input can produce several millions of output sentences. In order to find the best sentence a statistical ranker is introduced that chooses the most fluent sentence. To speed up this ranking process, the candidate sentences are packed into a lattice. This way, any sequence of words that appear in multiple sentences only needs to be ranked once. To learn the preferred expressions, a corpus of 46 million words of Wall Street Journal articles was used to create a bigram model. A Viterbi traversal of the lattice was performed to ultimately rank the sentences.

Although the NITROGEN system introduced a new way of performing sentence realization, the process of traversing is very slow and raises some computational problems that the HALogen system Langkilde (2000) tried to tackle. First of all, the lattices in NITROGEN were often too large to be traversed in an efficient manner. In HALogen this is resolved by packing whole sentences and sentence parts into a forest, leaving only the first and last words of each sub-tree to be considered if they occurred within other sentences. This way, when the sentences are generated, the system can omit the included trees completely, since all it needs are the words at the end. Moreover, in order to capture long-distance dependencies (because the lattice and the bi-grams cannot apprehend long-distance dependencies), HALogen incorporates additional syntactic information.

2.2.2 Syntactic features

Like NITROGEN and HALogen, the FERGUS generation system of Bangalore and Rambow (2000) also uses an n-gram language model. FERGUS, however, is expanded with a tree-based stochastic model and a wide-coverage grammar for English (XTAG) that is based on the lexicalized Tree Adjoining Grammar formalism. The generation process in FERGUS includes three components: a Tree Chooser, a tile Unraveler, and a Linear Precedence Chooser. The input to the FERGUS system is a dependency tree represented in a predicate-argument structure. The unordered nodes are only labelled with lexemes and contain no further syntactic annotations. The Tree Chooser provides these annotations by use of a stochastic tree model. It annotates the nodes syntactically in order to create derivation trees with some form of order. In addition, the Tree Chooser assumes that the choice of a tree for a node only depends on its daughter nodes. The model is trained on an annotated corpus of derivations from the grammar. Since there usually are different methods for attaching a daughter node to her mother, the output from the Tree Chooser does not completely define the surface string. After annotation, the grammar-based Unraveler maps all the possible realizations that are compatible with the trees to a word lattice that encodes the strings represented by each level of the derivation tree. In the final stage of the generation process in FERGUS, the Linear Precedence Chooser extracts the most plausible realization from the lattice on the basis of an n-gram language model. The n-gram statistics are derived from unannotated text from the Wall Street Journal corpus, similarly to the Nitrogen-system. In this case, however, a trigram model is used instead of a bigram model. The Dependency Structures used as input for FERGUS are fairly specified, and although FERGUS performs some lexical choice and syntactic choice, most of the compulsory decisions to be made are related to the order of the words.

The former systems used n-gram models based on the surface form to select the most fluent realization. In the EXERGE-system for Spanish-English (Habash (2004)) this principle is extended by using n-gram counts of words in dependency relations. In Habash (2004) the EXERGE module is described as situated somewhere in between the previously mentioned systems HALogen and FERGUS in terms of input complexity and the balance of statistical and symbolic components. In order to make lexical and structural choices, n-grams based on pairs of parent-child lexemes are derived from parsed sentences from the English UN corpus. The generation component of the EXTERGE-system consists of seven steps. In the first five steps, lexical and structural selection is performed while in the last two steps the sentences are realized. In the first steps of the generation process, after a dependency structure is given as input to the system, a rule-based component performs structural expansion and syntactic assignment to produce a forest of syntactic dependencies. The structural n-gram model is also used for expanding the syntactic structure of noun phrases in this forest. In the next step, a bottom-up algorithm applies the structural n-gram model in order to reduce ambiguous nodes in the forest. This pruning is done in order to decrease the size of the packed word forest that is created in the following stage of the realization process, using a rule-based generation grammar. Finally the ranking module of HALogen (Langkilde (2000)) is applied for the extraction of the best string using a standard linear bigram model.

2.2.3 Maximum entropy models with N-gram features and disambiguation features

Due to limitations of the previously mentioned sequential and surface-oriented models, a more linguistically informed approach of using models that are sensitive to the core structure of competing realizations could be beneficial. More sophisticated systems use a combination of n-gram language models and Maximum Entropy (MaxEnt) models. MaxEnt models are linear classifiers that can incorporate arbitrary features. Since feature-based models can contain more information, they are able to perform better than n-gram language models.

In the generation part of the LOGON system (Velldal et al. (2004), Velldal and Oepen (2005), Velldal and Oepen (2006a), Velldal and Oepen (2006b) realization ranking is implemented after the generation of candidate sentences from semantic specifications. Velldal et al. (2004) compared the performance of an n-gram model with a Maximum Entropy model from a parsing application trained on the small, domain-specific corpus. In order to create training data for the different models, they constructed a symmetric treebank composed of a set of pairings of surface forms and associated semantics, a set of alternative analysis for each surface form and a set of alternative realizations of each semantic form. The system produces multiple candidate sentences, which are ranked by three different models: an n-gram language model, a MaxEnt model using (HPSG) structural features and a combination of the two models. They found that the MaxEnt model performed better than the n-gram model but the combination of both yielded even better results. In addition to Velldal et al. (2004), Velldal and Oepen (2005) added more structural features and n-grams. This resulted in a substantial improvement in the performance of the MaxEnt ranker. Velldal and Oepen (2006a) compared the performance of an n-gram model, a Maximum Entropy model using structural information, including the language model as a separate feature, and a Support Vector Machine (SVM) ranker trained on the same feature set. They found that the SVM ranker and the MaxEnt ranker produced comparable results without significant differences. The MaxEnt model, however, needs far less training time and computational memory, which makes it a more practical model for the task of fluency ranking.

Building on Velldal and Oepen (2005), Cahill et al. (2007) tested a log-linear model for realizing German sentences with a Lexical Functional Grammar (LFG) generating sentences from so-called f-structures. The model incorporates relative order of subject and object; sentence length and language model scores. Like Velldal et al. (2004) they constructed a symmetric treebank for the creation of training data. Also in this system a language model was applied in combination with structural features. For the creation of structural features, instantiated templates were outlined. The feature templates included information from f-structure and c-structure, simple properties such as the number of times a particular category label occurs or compound features. They found that the contribution the structural features make to the quality of the output is slightly better in the case of a free word order language like German than it is in the case of English. The number of structures used is also much larger than the data used in Velldal and Oepen (2005), although the improvement over a baseline language model was small. De Kok et al. (2011) and De Kok (2013) describe fluency ranking in the generation module of Alpino performed by the same statistical model as for parse selection. This process is executed by storing the multiple options as partial representations and next combining them by use of a Maximum Entropy model. One important difference between this ranker and the one previously described, is the level of abstraction of the features for sentence construction.

Feature selection was applied and it was found that only a small number of features is required to rank the generated sentences effectively by fluency. A small number of features that model word and part-of-speech trigram distributions, topicalization, modifier adjoining, and ordering in the middle field were effective to describe the fluency of a sentence.

In their treelet system, Quirk et al. (2005) use a word order model trained as a decision tree that assigns probabilities to word orders of target trees given the source trees. This order model makes the assumption that the position of each child can be modelled independently in terms of its position relative to its head. Their features model whether a modifier is ordered to the left or right of its head, and how far away, with features containing information about word and part of speech of the head and modifier. In Menezes et al. (2006) this approach is adapted to a log-linear model with features chosen to maximize performance on a development set. The weights of the individual components in the model are set by an automatic method for parameter tuning. Chang and Toutanova (2007) introduced a global order model for English to Japanese translation. The model ranks n-best dependency tree output of the treelet system using local features that capture head-relative movement and global features that capture the surface movement of words in a sentence. Menezes and Quirk (2007) improved this method using a so-called "dependency order template" system that evades the massive amount of possible combinations of reordering treelets they encountered in their work in 2005, necessitating severe pruning of the search space. The order templates are unlexicalized transductions that map the dependency trees containing only parts of speech to unlexicalized target language trees. These order templates are extracted from dependency trees and word alignments of the training data. Ultimately, the order templates are combined with the relevant treelet translation pairs in order to construct lexicalized transduction rules.

3 Evaluation

The evaluation of the performance of each module and of the individual tasks of the generation process increasingly has been given more prominence in Natural Language Generation. The goal of the evaluation of both the coverage and the quality of a generation system, is to measure the extent to which a sentence can be represented and generated (Reiter and Belz (2009)). The results of a generation process are particularly challenging to evaluate because of the difficulty to automatically measure grammaticality and the fact that multiple outputs are possible.

3.1 Corpus-based evaluation

In recent years there has been growing interest in evaluating automatically generated texts by comparing them to a corpus of reference texts. In this setting automatic metrics such as string-edit distance, tree similarity, or BLEU (Papineni et al. (2002)) are used. Corpus-based evaluation has been specifically prevalent in the evaluation of surface realizers. A reason for this could be that the most important characteristic of many sentence realizers is grammatical coverage which can be evaluated well by robust treatment of special and unusual cases and corpus-based procedures (Reiter and Belz (2009)). Langkilde-Geary (2002), for example, used a section of the Penn Treebank to evaluate the coverage and quality of the HALogen system. First the input was automatically constructed from the Treebank annotation and then regenerated by the generation system. The output was

then compared to the original input sentence. Next to corpus-based evaluation also other validation methods have been studied. In recent years some validation studies have been carried out that studied correlations between automatic metrics and human evaluations. One of these studies is Bangalore et al. (2000), who considered string-edit and tree-edit metrics using a small number of manually simulated system outputs. It is also possible to evaluate the quality of the generation system on the basis of the output of a parser. In this case evaluation data can be created by parsing sentences. The dependency structure corresponding to the best parse as selected by the disambiguation component is then extracted and assumed to be the correct parse. It is then possible to assign a quality score to each realization by comparing it to the original sentence.

3.2 Metrics

Research in NLG has developed evaluation metrics based on the comparison of output texts with a corpus of human texts, and have shown that some of these metrics are highly correlated with human judgments (Reiter and Belz (2009)). Since there still is no common ground about what metric is best for the evaluation of the generation process, in this project three of them (BLUE, ROUGE, GTM) will be used and compared. Later in the project, we will examine the possibility of adapting more analytic measures for MT quality assessment (see the forthcoming Deliverable D3.3) for the task of evaluating the generation output.

The BLEU metric (Papineni et al. (2002)) calculates the number of n-grams a generated string shares with a reference string, adjusted by a brevity penalty. Usually the geometric mean for scores up to 4-grams are reported. BLEU scores range from 0 to 1, where 1 is the highest reachable score. This number, however, can only be reached if all its substrings can be located in one of the reference texts. It should, furthermore, be calculated on a large test set with several reference translations. This metric has been widely used in Machine Translation since properly calculated BLEU scores are believed to correlate reliably with human judgments. Callison-Burch et al. (2006), however, have found some evidence that BLEU may not correlate with human judgment to that extend that it is currently assumed to do. This could, for example, occur when the systems that are evaluated don‘t share the same lexicon since they are based on a different paradigm and BLEU limits its scope to the lexical dimension.

The ROUGE metric (Hovy et al. (1996)) is designed to evaluate automatically generated summaries and comprises a number of string comparison methods including n-gram matching. There are several different ROUGE metrics of which ROUGE-N is the most straightforward (Reiter and Belz (2009)). This metric computes the highest amount of n-grams that are matched in a reference summary and the generated summary. Subsequently, a method is applied that medians the score across leave-one-out subsets of the set of reference texts. ROUGE-N is an almost straightforward n-gram recall metric between two texts, and has several counter-intuitive properties, including that even a text composed entirely of sentences from reference texts cannot score 1. ROUGE-SUN on the other hand, looks at so-called ‘skip bigrams‘ that occur in both the generated and reference texts. A skip bigram consists of two words that are not necessarily adjacent, but may be divided by up by intermediary words.

The last metric that will be used in this project is GTM (General Text Matching, Melamed et al. (2003)). This metric calculates the word overlap between a reference and a solution without counting duplicate words. This metric places less importance on word

order than the BLUE metric does. Cahill (2009) examined various metrics in the context of a sentence generation system for German and compared them with human judgements and found that the General Text Matcher had the highest correlation.

3.3 Task-based evaluation

Task-based or extrinsic evaluation can be used in real usage scenarios where the text is used by humans to make decisions or perform actions. In this evaluation method the generated text is given to a person that assesses how well it helps him or her perform a task Reiter and Belz (2009). In the QTleap project, for example, a system which answers to computer-related problems can be evaluated by giving the answers to users, and let them asses whether the answers helps them solving the problem. Depending on the study design, these studies often don‘t answer the question of which aspects of a system contribute to its success or failure. Also, although task-based evaluations have traditionally been regarded as a good evaluation method in NLG, they are time-consuming and expensive, and can be difficult to carry out.

4 Beyond the State of the Art

In order to improve the state of the art for deep machine translation, various improvements and innovations are foreseen for the generation components of the various languages involved in this task.

4.1 Realization based on deep representations

In order to be able to generate from the deep semantic representations, as proposed in the project (cf. deliverable 4.1 on deep semantic processing), there are various tasks, which need to be taken into account. The amount of work for this task differs per language, and is dependent on the actual status of the generation component of that language, both in terms of the level of maturity of that component, as well as the nature of the input representation currently assumed for generation. For instance, for Dutch we will employ the Alpino system, which does contain a generation system. However, in order to make the Alpino generator suitable for machine translation it is foreseen that a number of problems have to be solved. On the one hand, the current representation that the generator assumes as input (abstract dependency structures) is not abstract enough for the purposes of translation. In terms of dependency grammar, as advocated by researchers from Prague, we foresee that the input to generation is closer to a tectogrammatical representation (e.g., with semantic roles rather than grammatical functions) whereas the current representation is closer to analytical representation. On the other hand, the current generation component has been developed from a monolingual point of view, and has never been seriously tested against real-world input. It is expected that the application of the generator to new, previously unseen, inputs will give rise to subtle adaptations and improvements of the generation component. Some of these adaptations include the treatment of capitalization and punctuation, but undoubtedly, further as yet unexpected technical problems will surface.

Since there is no generation component for Bulgarian, IICT-BAS plans to try generation from MRS (RMRS) to text. For the direction English-to-Bulgarian we will experiment with the following setup. The English MRS structure will be transferred to a

Bulgarian MRS. From the Bulgarian MRS we will generate Bulgarian sentence containing lemmas annotated with POS information from MRS. The POS information will be distributed over the sentence via various mechanisms, such as agreement. Then a module for morphological generation will be applied in order to generate the actual word forms. The transfer between English MRS to Bulgarian MRS will be done along the lines of LOGON project. Some transfer rules will be learned from the English-Bulgarian lexicon. Other (together with appropriate context) will be learned from automatically processed parallel corpora. For the generation of Bulgarian sentences we will exploit a large monolingual corpus automatically annotated with MRS structures.

Another approach we would like to experiment with, is to adapt the approach of Hidden Tree Markov Models (Žabokrtský and Popel (2009)) to MRS structures where the dependency trees are substituted with the appropriate representation of MRS structures. For the direction Bulgarian-to-English we will experiment with similar to the above approaches. Additionally we will exploit the English Resource Grammar (ERG) in the generation mode directly from MRS structures. The challenge here is that ERG needs well-specified MRS structures in order to be able to generate the corresponding sentences. Thus, the main problem will be for the transfer component to create such MRS structures.

In the German hybrid system used for Pilot 1, generation from deep structures is performed by the Lucy generation component. Issues observed on the usage data of the QTLeap corpus are mostly results of parsing and/or transfer errors. A frequent issue when translating into German are cases where English items as in "Go to menu Layer > New > Layer [...]" are partially generated as compounds in German as in "Gehen Sie zu Menü-Schicht > Neue >-Schicht [...]". Sometimes, this compounding even crosses the boundaries given by the ">" signs that are correctly interpreted as punctuation. In Pilot 1, the linear combination of Lucy and Moses is able to take care of some of the issues. A solution is to fix the treatment of these structured noun lists in Lucy. In the TectoMT system for German that is still work in progress, generation will be handled by the Zmorg finite-state transducer that has been chosen to match the output of the ParZu parser. Issues might occur if it turns out that the result of the Tecto transfer component cannot unambiguously be interpreted by the generator. Additionally, generating compounds like the ones mentioned above may also require some effort.

4.2 Robustness

One particular problem that will arise for generation systems if they are applied in a machine translation context is robustness. For grammar based systems, such as the generator included in the Alpino system, it may happen that the input structure for generation is such, that - according to the grammar - there is no target language sentence for that input structure. One simple example of this scenario arises if the inputs structure contains (a representation of) a word, which is not known to the target grammar and dictionary. More complicated examples arise, if the grammar cannot generate a result because of certain structural properties of the input structure. In QTLeap, we may be somewhat optimistic that this situation does not arise too often, because we assume that our transfer models are trained on parallel treebanks. However, we cannot safely assume that this situation will never occur. Therefore, an important task consists of the inclusion of new robustness techniques, and the extension of existing robustness techniques. At least two techniques will be implemented. First of all, techniques that treat unknown (representations of) words are important. Such techniques will ensure that a suitable

target syntactic representation of the unseen word is generated, which entails that the generation procedure of the full sentence can be completed - perhaps by using the source language orthography of the unseen word in the target sentence. Secondly, the generation procedure should be (re-)designed in such a way, that, if the input specification does not lead to a single candidate sentence, the generation component produces phrases for the various parts of an input representation - which can then be combined in target sentence fragments.

4.3 Multi-word expressions

Further improvements in the generation component are foreseen with respect to the treatment of multi-word expression. Given the heterogeneous nature of the phenomena that usually is included under this rubric, the repercussions for a generation component are different. On the one hand, some multi-word expressions will be dealt with in the lexicon and grammar. Some others will require explicit attention by a generator. The goal in QTLeap for multi-word expressions, is to treat these in a deep, semantically oriented, way. This implies that from the point of view of transfer, multi-word expressions may not require special techniques: the multi-word expressions are mapped to a simple source language predicate. That predicate is mapped to a target language predicate. Finally, that target language predicate is then input to the generator. For fixed and semi-fixed expressions, we may assume that the target language grammar and dictionary already treats the expressions in a satisfactory way. For more flexible expressions, it may be that pre-processing to the input structure is required, before the generator is applied.

4.4 Linked open data/Wordnet

As proposed in the 4.1 deliverable, and consistent with work package 5, the interface representations for transfer will incorporate information from Linked Open Data and/or Wordnet synsets. In the analysis direction, this implies that an alignment between source language dictionaries and the linked resources is assumed, in such a way that a particular word is mapped (using word disambiguation) to a node in a linked inventory. Lexical choice in generation will then be the problem to decide which actual word form is to be used for a given node. To make this concrete, we consider the use of Euro-Wordnet. In analysis, each word is mapped to a Wordnet synset. In generation, the task is to generate actual words on the basis of a Wordnet synset. An interesting experiment can be performed to answer the question whether context-based generation of such synset identifiers out-performs a baseline system in which simply the most frequent word in a given synset is selected.

We furthermore plan to go beyond Wordnets and semantic lexicons, exploiting the knowledge within the web LOD, such as DBpedia, Geonames, etc. We will map the facts and relations to the semantic and valency lexicons as well as to the NE lists. In this way, the sense disambiguation step will be facilitated. Here we will experiment with exploiting conceptual knowledge in order to support a better transfer on the level of MRS structures and in the generation process itself. Two cases will be considered:

- Instance data incorporation, and
- Terminological data incorporation.

The first one will be connected to the generation of Named Entities for entities recognized in the source language. The names for these entities will be extracted from LOD datasets such as DBPedia, GeoNames, etc. The terminological conceptual knowledge includes classes and properties of the things mentioned in the source text. Their inclusion into the MRS structure will provide better mechanisms of transfer and a better approach to generation, since their conceptual information will facilitate generation when there is no specific information within the generation grammar.

It would moreover make sense to try and either improve the generation components of RBMT and SMT (or hybrid) systems or to devise post-fixing mechanisms as counterbalance to the end-to-end deep approach. This would maximize the impact of results, e.g, of the principled treatment of multi-word expressions or the inclusion of LOD if they could be integrated also into proven processing chains while minimizing the potential risk that the full deep approach is not manageable or does not yield the expected performance (see also Task 2.3).

5 Expected benefits for the real usage scenario

Within the project, fresh data is produced continuously in form of a real usage (IT-helpdesk) scenario, provided by the partner HF. This QTLeap corpus is characterized by short sentences, usually a request of help followed by an answer. The request for help is often a well-formed question or a declarative sentence reporting a problem, but in a relevant number of cases, the question is not grammatically correct, presenting problems with concordance, missing verbs, etc. In some cases, the request is composed by a list of key words. This kind of utterance is representative of informal communication via chats. On the other hand, a more formal register characterizes the answers, as they are produced by well-trained operators and they need to be very precise and concise in order to clarify the user request and to not generate more doubts. In the next sections some example sentences from the baseline translations are shown and some suggestions are described about how improving the generation component could improve them. A baseline pilot 0 was created and applied on a subset of this corpus. These Moses systems (Koehn et al. (2007)) were set up with a basic phrase-based setup (see deliverable 2.2). When reviewing the output sentences for both translation directions, some serious errors can be found that have better chances to be solvable by a generation component advanced along the research path pursued here.

5.1 Fluency

As mentioned earlier in this deliverable, fluency is an important aspect of the generation component in MT. In the output data of pilot 0, a great deal of non-fluent sentences occur. A good example is the first sentence in table 1 which has been translated from English to Dutch. Although the sentence contains all the necessary words for a good sentence, the order of the words is not correct. When using some imagination, the sentence is still understandable but it takes more effort to grasp the correct meaning. The aim in this project is to apply an advanced fluency ranker, as described in section 2.2.2, in order to obtain more fluent, and therefore more understandable, sentences. Since the generated questions will not be consumed by humans, but rather by the matcher in a QA system, fluency is not a big issue in the opposite translation direction.

EN=>NL	
Original	As you type, the document is saved automatically.
Output:	Terwijl u typt, het document opgeslagen wordt automatisch.
Preferred output:	Terwijl je typt, wordt het document automatisch opgeslagen.

Table 1: Influent translations

5.2 Choice of words

Another frequent error that occurs in the translated sentences from Pilot 0, is a wrong choice of words. Some examples of these errors can be seen in table 2. In the first sentence, translated from English to Dutch, the noun *ports* is translated with **haven**. In other contexts, for example within a navigation domain, this would have been a good translation. Here, however, the preferred translation is *poorten*. This confusion not only results in a strange sentence that does not convey the same meaning as the original one. The matcher of the question and answering system could therefore have serious issues finding the write answer due to the erroneous meaning of the translated word. It would search for a *harbour* instead of a *port on a computer*. Something similar happens in the second example. Here only the first part of the expression *junk mail* is translated with the Dutch word for *rag* or *piece of junk*. Here the end user would get an answer containing a completely different meaning which would be hard to understand and solve the problem. The choice of wrong words also occurs in the other translation direction. In the first example, the word *stof* has multiple meanings in Dutch. It could mean *dust* with the meaning of *material made of fibers* which is the preferred translation in this case. In English, however it could also mean *substance* in contexts of *anything that has mass and takes up space*.

Errors like these could possibly be avoided by implementing Wordnets synsets into the generator in order to choose an acceptable synonym and generate a sentence that has the same meaning as it's original one. As mentioned in the last chapter, the interface representations for transfer will incorporate information from WordNet and/or Linked Open Data. In the analysis direction, this implies that an alignment between source language dictionaries and the ontological resources is assumed, in such a way that a particular word is mapped to a node, by use of word sense disambiguation, in an LOD or WordNet inventory. This way, in the generation direction, the intended sense and the expected output word to be generated is clear. Also the wrong translation of Named Entities could be avoided by generating from LOD resources. By incorporating Instance data and connect it to the generation of Named Entities for entities recognized in the source language. Also, the quality of the translated sentences could benefit from the addition of more syntactic information in the generation component like, for example, the use of Part-of-Speech tags combined with dependency structures in both the analysis as the generation component.

6 Conclusion

In this deliverable, previous work on the task of the generation of target language texts from deep linguistic analysis has been evaluated. First, the implementation of algorithms that construct sentences that are consistent with a given deep structure has been discussed. Subsequently, the development of statistical language modeling components that

En=>NL	
Original	In Linux, how can I see all the **ports** that are open on the system?
Output:	In Linux, hoe kan ik zie alle de **havens** die openstaan op het systeem?
Preferred output:	In Linux, hoe kan ik alle **poorten** zien die open zijn in het systeem?
Original	Spam is an internet version of **junk mail**.
Output:	Ongewenstmessage status is een versie van **vodden**.
Preferred output:	Spam is een internet versie van **ongewenste post**.
NL=>EN	
Original	**Stof** maakt het koelen van de computer moeilijker en in extreme gevallen kan dit leiden tot schade aan de hardware.
Output:	**Substance** is products cools the computer difficult and in extreme cases, this can lead to damage to the hardware.
Preferred output:	**Dust** makes the computer cooling more difficult and in extreme cases, it may lead to hardware damage.
Original	Klik op Account en klik waar Volgers **staat.**
Output:	Click on Account and click where volgers **state.**
Preferred output:	Click the tab that says , "Account", then press where **it says** followers.

Table 2: Wrong choice of words

ensure that the most fluent candidate sentence is selected has been described. Then, evaluation methods and metrics are reviewed. Finally, goals for further improvement of the generation components have been emphasized and their benefits for the real usage scenario are reported.

References

Ambati, V. (2008). Dependency structure trees in syntax based machine translation. In *Adv. MT Seminar Course Report*.

Bangalore, S. and Rambow, O. (2000). Corpus-based lexical choice in natural language generation. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, ACL '00, pages 464–471, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bangalore, S., Rambow, O., and Whittaker, S. (2000). Evaluation metrics for generation. In *Proceedings of the First International Conference on Natural Language Generation - Volume 14*, INLG '00, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.

Cahill, A. (2009). Correlating human and automatic evaluation of a German surface realiser. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort '09, pages 97–100, Stroudsburg, PA, USA. Association for Computational Linguistics.

Cahill, A., Forst, M., and Rohrer, C. (2007). Stochastic realisation ranking for a free word order language. In *Proceedings of the Eleventh European Workshop on Natural Language Generation*, ENLG '07, pages 17–24, Stroudsburg, PA, USA. Association for Computational Linguistics.

Callison-Burch, C., Osborne, M., and Koehn, P. (2006). Re-evaluation the role of BLEU in machine translation research. In McCarthy, D. and Wintner, S., editors, *EACL*. The Association for Computer Linguistics.

Chang, P.-C. and Toutanova, K. (2007). A discriminative syntactic word order model for machine translation. In Carroll, J. A., van den Bosch, A., and Zaenen, A., editors, *ACL*. The Association for Computational Linguistics.

Copestake, A. (2002). *Implementing typed feature structure grammars*. CSLI lecture notes. CSLI Publications, Stanford, CA.

Copestake, A., Flickinger, D., Malouf, R., Riehemann, S., and Sag, I. (1995). Translation using minimal recursion semantics. In *Proc. TMI95*, Leuven, Belgium.

Copestake, A., Flickinger, D., Pollard, C., and Sag, I. (2006). Minimal Recursion Semantics: An Introduction. *Research on Language & Computation*, 3(4):281–332.

De Kok, D. (2013). *Reversible Stochastic Attribute-value Grammars*. Groningen dissertations in linguistics.

De Kok, D., Plank, B., and van Noord, G. (2011). Reversible stochastic attribute-value grammars. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 194–199, Stroudsburg, PA, USA. Association for Computational Linguistics.

Habash, N. (2004). The use of a structural n-gram language model in generation-heavy hybrid machine translation. In Belz, A., Evans, R., and Piwek, P., editors, *INLG*, volume 3123 of *Lecture Notes in Computer Science*, pages 61–69. Springer.

Hays, D. G. (1964). Dependency theory: a formalism and some observations. *Language*, 40:511–525.

Hovy, E., van Noord, G., Neumann, G., and Bateman, J. (1996). Language generation. *Survey of the State of the art in Human Language Technology*.

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., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

Langkilde, I. (2000). Forest-based statistical sentence generation. In *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, NAACL 2000, pages 170–177, Stroudsburg, PA, USA. Association for Computational Linguistics.

Langkilde, I. and Knight, K. (1998). Generation that exploits corpus-based statistical knowledge. In *Proceedings of the 17th International Conference on Computational Linguistics - Volume 1*, COLING '98, pages 704–710, Stroudsburg, PA, USA. Association for Computational Linguistics.

Langkilde-Geary, I. (2002). An empirical verification of coverage and correctness for a general-purpose sentence generator. In *Proceedings of the International Natural Language Generation Conference*.

Lin, D. (2004). A path-based transfer model for machine translation. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA. Association for Computational Linguistics.

Melamed, I. D., Green, R., and Turian, J. P. (2003). Precision and recall of machine translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Companion Volume of the Proceedings of HLT-NAACL 2003–short Papers - Volume 2*, NAACL-Short '03, pages 61–63, Stroudsburg, PA, USA. Association for Computational Linguistics.

Menezes, A. and Quirk, C. (2007). Using dependency order templates to improve generality in translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 1–8, Prague, Czech Republic. Association for Computational Linguistics.

Menezes, A., Toutanova, K., and Quirk, C. (2006). Microsoft research treelet translation system: Naacl 2006 europarl evaluation. In *Proceedings of the Workshop on Statistical Machine Translation*, StatMT '06, pages 158–161, Stroudsburg, PA, USA. Association for Computational Linguistics.

Oepen, S., Dyvik, H., Lønning, J. T., Velldal, E., Beermann, D., Carroll, J., Flickinger, D., Hellan, L., Johannessen, J. B., and Meurer, P. (2004). Som å kapp-ete med trollet? - towards mrs-based norwegian-english machine translation. In *In Proceedings of the 10th International Conference on Theoretical and Methodological Issues in Machine Translation*, pages 11–20.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Quirk, C., Menezes, A., and Cherry, C. (2005). Dependency treelet translation: Syntactically informed phrasal smt. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 271–279, Stroudsburg, PA, USA. Association for Computational Linguistics.

Reiter, E. and Belz, A. (2009). An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Comput. Linguist.*, 35(4):529–558.

Reiter, E. and Dale, R. (2000). *Building Natural Language Generation Systems*. Cambridge University Press, New York, NY, USA.

van Noord, G. (2006). **A**t **L**ast **P**arsing **I**s **N**ow **O**perational. In *TALN 2006 Verbum Ex Machina, Actes De La 13e Conference sur Le Traitement Automatique des Langues naturelles*, pages 20–42, Leuven.

Velldal, E. and Oepen, S. (2005). Maximum entropy models for realization ranking. In *In Proceedings of the 10th Machine Translation Summit pp. 109*, page 116.

Velldal, E. and Oepen, S. (2006a). Statistical ranking in tactical generation. In *Proceedings of EMNLP 2006*.

Velldal, E. and Oepen, S. (2006b). Statistical ranking in tactical generation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 517–525, Stroudsburg, PA, USA. Association for Computational Linguistics.

Velldal, E., Oepen, S., and Flickinger, D. (2004). Paraphrasing treebanks for stochastic realization ranking. In *Proceedings of the 3rd Workshop on Treebanks and Linguistic Theories*.

Žabokrtský, Z. and Popel, M. (2009). Hidden Markov tree model in dependency-based machine translation. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort '09, pages 145–148, Stroudsburg, PA, USA. Association for Computational Linguistics.