The effect of a few rules on a data-driven MT system

Vincent Vandeghinste, Peter Dirix, Ineke Schuurman
Centre for Computational Linguistics
KULeuven
vincent,peter,ineke@ccl.kuleuven.be

Abstract

In this paper, we test the METIS-II MT system, from Dutch to English, under several experimental conditions: a verbatim condition in which word by word dictionary translations are used, a condition in which the effect of adding a target language corpus lookup is measured, and the effect of adding a few transfer rules to this. The results indicate that the addition of transfer rules clearly improves our system, but we find somewhat surprising results in the verbatim condition.

1 Introduction

The METIS-II system[1] is a corpus-based machine translation system, which does not make use of a parallel corpus, but instead uses shallow source language analysis, a dictionary, some mapping rules, and a target language corpus.

The METIS-II baseline approach to machine translation from Dutch to English consists of the following steps:

As our *source-language model*, we use a shallow source language analysis, which consists of the following steps:

- tokenisation,
- PoS-tagging, using TnT (Brants, 2000), with the D-CoI tag set (Van Eynde, 2005), allowing several tag alternatives, each with their own weight,
- lemmatisation, through lexicon lookup using the CGN-lexicon (Piepenbrock, 2002),
- chunking, using ShaRPa2.0 (Vandeghinste, 2003, 2005), including head detection,
- clause detection.

For the *translation model*, the mapping between the source-language tag set and the target-language tag set is done by a many-to-many mapping of the source-language tag set onto the target-language tag set (CLAWS5[2]), and the use of a bilingual dictionary that is lemma-based. The effect of the addition of some transfer rules (which also belong to the translation model) is investigated in this paper.

The basic unit used in the METIS-II approach is the lemma, as this reduces the number of dictionary entries, as well as the data sparsity.

Several translation candidate bags are generated, each consisting of lower-level bags, with each bag representing a source-language chunk, clause or sentence. A bag is an unordered list.

In the *target-language model*, these bags are matched bottom up with a target language corpus (in our case, the BNC), which results in target-language chunks, clauses, or sentences with a weight, based on how well they match with the corpus, allowing us to order several translation candidates for each source-language part.

For this, we preprocessed the BNC by lemmatizing it with a reversible lemmatizer (Carl, Schmidt, and Schütz, 2005), chunking it with ShaRPa2.0, and detecting the clauses with a rule-based clause detector. The resulting chunks are put in a psql[3] database, organised by chunk type, and indexed on the heads, to allow fast information retrieval.

[1]Supported by the 6th European Framework Programme, FP6-IST-003768

[2]http://www.comp.lancs.ac.uk/ucrel/claws5tags.html

[3]http://www.postgres.org

In a final stage, the target-language tokens are generated, based on the translated lemmas and mapped tags (Carl et al., 2005).

For more details on this approach, see Dirix, Vandeghinste, and Schuurman (2007).

This approach works fine for noun phrases and prepositional phrases, even allowing us to select the correct translation candidate amongst several possibilities offered by the bilingual dictionary.

Although for related languages, word-to-word translations could give a reasonable outcome (cf section 3), this is not always true. Dutch has a rather specific sentence structure, especially in subclauses. Since we only use shallow analysis, there are too many parts of a sentence to be ordered and selected from amongst a set of alternatives. Furthermore, the probability of finding a match in the corpus is reduced because of the large number of lemmas that need to be found in the same sentence or clause. Sentence and clause level structures are less likely to be the same in Dutch and in English than noun and prepositional phrases.

To map the sentence or clause structure of the source language onto the sentence or clause structure of the target language we introduced a number of transfer rules to the *translation model*. It is the effect of these transfer rules we describe in this paper.

2 Transfer rules

The following transfer rules were introduced to better map the source-language sentence structure to the target-language sentence structure.

These rules are source-language-dependent, but are implemented as such that, for a different source language with the same phenomena as the current source language, the rules should be triggered as well. This means that the conditions on which the rules fire are using properties in terms of target language words (the translation of the Dutch trigger words from the dictionary) and target-language tags.

2.1 Verb-group treatment

What we consider a verb group is any sequence of verbs, possibly with a negation inbetween. This is a useful way of chunking for tenses like the present perfect, which consist of an auxiliary and a past participle. In Dutch, the auxiliary and the past participle can be separated, while in English they stay together (Huddleston and Pullum, 2002). So, after each word from our source sentence has been translated, but our sentence is still in the source language structure, we detect if, within the same clause, we find an auxiliary and a past participle, so we can put them in the same verb group. This makes sure that in our target sentence, they stay together, as words belonging to the same chunk should never be separated by the target language model.

Example	
Dutch	*Hij heeft de wedstrijd gewonnen.*
lit.	`He has the game won.`
English	*He has won the game.*

The resulting verb group bag contains the auxiliary and past participle, which are then put in target-language order, using the target-language corpus lookup, in the same way as it is used for e.g. NP matching.

2.2 The Dutch tense system[4]

As in most West Germanic languages, the Dutch indicative only has two non-compound tenses: the present and the imperfect active tense. The subjunctive is only used in a few archaic expressions and is not considered here. The imperative and the infinitive only have one, non-compound form.

All other tenses of the indicative are composed of an auxiliary and the past participle or the infinitive. For the future tense and the conditional tense, it is simple: the present resp. imperfect tense of the verb *zullen*, followed by the infinitive. The passive voice of these four tenses is formed, of course only for transitive verbs, by the appropriate forms of *worden*, combined with the past participle.

The perfect tenses are generally formed with *hebben* and the past participle in the active voice and *zijn* and the past participle in the passive voice. An example of all 1st persons singular of these type of verbs is shown in tables 1 and 2. However, some intransitive verbs, esp. verbs that change the state of the subject, have an active form with *zijn* (cf table 3). Some verbs, e.g. *ophouden* can have both, while verbs of motion have *zijn* or *hebben* depending on the context and the meaning.

[4]For an elaborated overview, see e.g. Haeseryn, Romijn, Geerts, De Rooij, and Van den Toorn (1997)

Table 1: Active tenses of *slaan* (to hit)

present	ik sla
imperfect	ik sloeg
future	ik zal slaan
conditional	ik zou slaan
present perfect	ik heb geslagen
past perfect	ik had geslagen
future perfect	ik zal geslagen hebben
conditional perfect	ik zou geslagen hebben

Table 2: Passive tenses of *slaan* (to hit)

present	ik word geslagen
imperfect	ik werd geslagen
future	ik zal geslagen worden
conditional	ik zou geslagen worden
present perfect	ik ben geslagen
past perfect	ik was geslagen
future perfect	ik zal geslagen zijn
conditional perfect	ik zou geslagen zijn

Table 3: Active tenses of *groeien* (to grow [intransitive])

present	ik groei
imperfect	ik groeide
future	ik zal groeien
conditional	ik zou groeien
present perfect	ik ben gegroeid
past perfect	ik was gegroeid
future perfect	ik zal gegroeid zijn
conditional perfect	ik zou gegroeid zijn

2.3 Tense mapping

When translating e.g. a present perfect, translating the words which form this tense does not result in an accurate translation. The correct translation depends on a number of factors. The rules performing these transformations are described in the following sections.

Note that the rules mentioned in this section are not an exhaustive list of the tense mapping phenomena which are required for optimal translation quality, but they can be considered as a first step towards bridging the gap between the source language and the target language.

2.3.1 *zijn* + past participle

In Dutch, there are two auxiliaries which can be used to form an active present or past perfect, *zijn* [E: to be] and *hebben* [E: to have], depending on the past participle, while in English only *have* can be used in the active form, while *be* is used in the passive form.

In Dutch, a past participle that combines with *zijn* can also be the passive form, when in the active form the past paticiple combines with *hebben*.

So, after translation of the words in the verb group, we look for the occurrence of *to be* (in finite form) combined with a past participle. As we do not know whether it concerns an active or passive form in the source language, we do not remove the translation candidate *be*, but we add two additional translation candidates for the verb group bag: in the first additional candidate we replace the auxiliary *to be* with the auxiliary *to have*; while in the second additional candidate we generate the auxiliaries for a present or past perfect continuous, *to have* in finite form and *be* in its past participle form.

Examples	
Dutch	*De trein <u>is</u> vertrokken*
lit.	`The train` <u>`is`</u> `left`
English	*The train <u>has</u> left*
Dutch	*Het onderzoek <u>is</u> tegen hem gericht*
lit.	`The investigation` <u>`is`</u> `against him aimed`
English	*The investigation <u>is</u> aimed against him*
Dutch	*Het boek <u>is</u> gevonden*
lit.	`The book` <u>`is`</u> `found`
English	*The book <u>has been</u> found*

It is up to the *target language model* to decide which of these three alternatives gets the highest weight, according to the frequency of occurence in the target language corpus.

2.3.2 *hebben* + past participle

In Dutch, the auxiliary *hebben* is used in active form only, and is translated by the dictionary into *to have*, which is the correct auxiliary in English active present or past perfect tense, so no transformations apply.

2.3.3 *worden* + past participle

In Dutch, the auxiliary *worden* is used in passive form only, while the dictionary lookup will translate *worden* into *to get*. So we need an extra translation candidate, when it combines with a past participle. A verb group bag containing the auxiliary *to be* and the past participle is generated, with the tense, number and person features of *worden*. This is then matched with the corpus.

In the following examples, the first one shows the use of *worden* as a regular non-auxiliary verb, while in the second example *worden* is the auxiliary for the passive form.

Examples	
Dutch	*Het wordt donker*
lit.	`It gets dark`
English	*It gets dark*
Dutch	*Hij wordt genoemd als de nieuwe premier*
lit.	`He gets named as the new prime minister`
English	*He is named as the new prime minister*

It is, again, up to the *target language model* to select the correct alternative, based on the target language corpus.

2.3.4 Continuous tenses

In Dutch, the continuous tenses are expressed with the appropriate form of *zijn*, followed by *aan het* + infinitive, e.g.:

Example	
Dutch	*Ik ben aan het rennen.*
lit.	`I am at the run`.
English	*I am running.*

Replacing *at the <VVI>*[5] with a continuous tense has not yet been implemented.

The Dutch continuous tenses are not used nearly as much as the English ones. This means a lot of the simple Dutch tenses have to be translated into continuous tenses in English, e.g.

[5]VVI is the CLAWS5 tag for a verb in the infinitive

Example	
Dutch	*Terwijl ik door de straat liep,...*
lit.	`While I through the street walked`, ...
English	*While I was walking through the street,...*

The solution is to add a mapping rule for the continuous form next to each simple form for all verb mapping rules. It is up to the target language corpus to select the appropriate form. This has not been implemented yet.

2.3.5 *hebben / zijn* + infinitive + infinitive

In Dutch, like in German, a temporal auxiliary (*hebben* or *zijn*) which is usually followed by a past participle, can come with an infinitive instead, when a third verb comes into play. This phenomenon is known as Infinitivus Pro Participio (IPP). In English, however, a temporal auxiliary should always be followed by a past participle.

Examples	
Dutch	Ik *heb* hem *proberen te bellen*
lit.	`I have him try to call`
English	*I have tried to call him*

We will implement a rule that will be triggered when a temporal auxiliary is followed by at least two infinitives. In that case, the first infinitive will be realised as a past participle, and no extra translation candidate will be generated.

2.4 OTI Mapping

In Dutch, there is a construction we call the *om te*-infinitive (OTI). It is a verbal projection, which is introduced by the preposition *om*. The construction contains a *te*-infinitive, which is chunked as such in our source language, and which can be translated as *to + infinitive*, but the introducing *om* is not translated.

Example	
Dutch	*Hij speelt om de wedstrijd te winnen.*
lit.	`He plays Ø the game to win`
English	*He plays to win the game.*

In this case, the rules do not add an extra translation candidate, but they modify the existing translation candidate.

2.5 Mapping towards *like to* + infinitive

In Dutch, we use the adverb *graag* to express the same idea as the English verb *like*. We map the verb which is modified by *graag* onto the infinitive following *like to*, and we map the tense, number, and person of that verb onto the English *like*. To trigger this rule, we added the translation *like to* <*VVI*> to the entry *graag* in our dictionary.

Example	
Dutch	*Hij zwemt graag.*
lit.	`He swims willingly`.
English	*He likes to swim.*

In this case, we add a translation candidate, and leave it up to the target language corpus to decide which translation gets the highest weight.

2.6 *do* insertion

When expression a negation in English, the verb *do* is inserted, combined with the negator *not*, for nearly all verbs. Only the auxiliaries and the modal verbs (e.g. *will, shall, can, must, may, ought*) can be negated without inserting *do*.

In Dutch every verb can be negated by the adverb *niet*. So to avoid generating a word by word translation, we insert *do* into the verb group, taking the tense, person and number from the Dutch verb, while the translation of that Dutch verb is put in the infinitive.

Example	
Dutch	*Ik zie hem niet*
lit.	`I see him not`
English	*I do not see him*

In this case, we do not add an translation candidate, but modify the current translation candidate.

do insertion is also necessary in the case of questions and imperatives, but this is not yet implemented.

3 The experiment

We wanted to know what the effect was of adding these *transfer rules*, so we performed an experiment on our system, to see if the addition of these rules enhances the BLEU (Papineni, Roukos, Ward, Zhu, 2001) and NIST (Doddington, 2002) scores.

3.1 Methodology

All experimental results are the outcome of sending the test set in batch mode to our translation system, with a beam width of 20. The beam width is the cut off point: when there are more than 20 alternatives, the system ranks them, according to their weight, and cuts off the list at the first alternative with a weight lower than the 20th alternative.

3.1.1 The test set

We have a test set of 50 Dutch sentences, selected from newspaper texts, with three human reference translations. These sentences are selected as such that they contain a number of classical difficult MT issues, of which a non-exhaustive list can be found in Vandeghinste, Schuurman, Carl, Markantonatou and Badia (2006).

This test set cannot be considered as a representative test set, as it is too small. But as it covers several difficult MT issues, it can be considered as somewhat similar to the test point method for automated machine translation evaluation, as described by Yu (1993). The resulting BLEU and NIST scores cannot be compared to the scores of other machine translation systems, but they can be compared to the scores by the same system on the same test set.

3.1.2 Experimental conditions

We have three experimental conditions:

The 'verbatim' condition In the verbatim condition, source language analysis is performed, and the tag mapping rules and dictionary are used. No use is made of the *target-language model* as we did not use the target-language corpus. Only part of the *translation model* is used as we did not use any transfer rules. This results in the generation of lots of translations: a combination of all the translations coming from the dictionary, in the source-language order. As we did not use the target language corpus, these translations all receive the same weight.

The 'no-rules' condition In this condition, we test the accuracy of our system, when using source language analysis, the tag mapping rules and the dictionary, and bags are matched with the target language corpus to allow lexical selection and word reordering.

The ‘transfer’ condition This condition tests the system under the same conditions as the ’no-rules’ condition, but with the transfer rules, as described in section 2

3.2 Results

We calculated BLEU and NIST scores for all three conditions. Our system generates several translation alternatives (dependent on the beam size, which is 20 for all tests described in this paper), each with a weight. The *top-weight* translations are those translations that receive the highest weight.

As our system is not always capable of generating only one best translation, we present two types of results:

The *average* results are the average BLEU and NIST scores of all the top-weight translations generated for that test sentence under that condition.

The *best* results are the highest BLEU and NIST scores of all the top-weight translations generated for that test sentence under that condition. These can be considered as the highest possible scores we could achieve for a sentence, when all lexical selection would be perfect.

The difference between the *average* and *best* scores is largest in the ‘verbatim’ condition. This is due to the fact that many more top-weight translation alternatives are generated under this condition, as our weighting system uses target language corpus information to perform lexical selection and word reordering. In the ‘verbatim’ condition, as there is no lexical selection, all word translations that are generated by the dictionary receive the same weight. As this leads to a combinatorial explosion, a maximum of 1000 translation alternatives was investigated.

The average number of top-weight translations in the ‘verbatim’ condition was more than 377 translations per test sentence (with a maximum of 1000 translations per test sentence!), while in the ‘no-rules’ condition, the average was 2, and in the ‘transfer’ condition, the average was 2.5 top-weight translations per test sentence.

Note that in the figures, you can see the scores for all fifty test sentences, while the average score of the test set is in bold and dashed.

3.2.1 Average BLEU and NIST scores

The results for the average BLEU and NIST scores, as shown in table 4 and figures 1 and 2 reveal the progress made by our system.

Figure 1: Average BLEU Scores

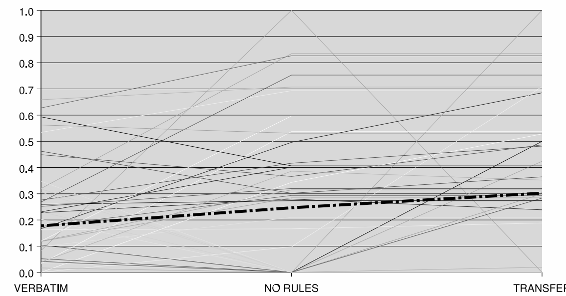


Figure 2: Average NIST Scores

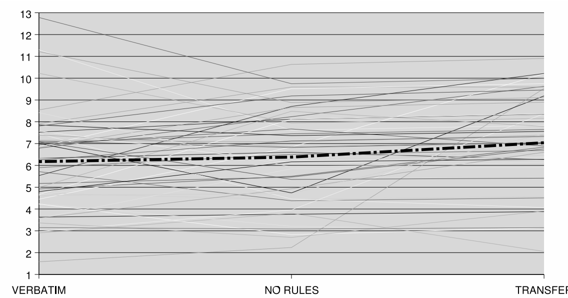


Table 4: Average BLEU and NIST scores for the three conditions

	verbatim	no-rules	transfer
BLEU	0.1776	0.2466	0.3024
NIST	6.1737	6.3800	7.0393

From ‘verbatim’ to ‘transfer’ a relative rise in BLEU score of more than 41% was reached. In NIST score this rise was more than 12% relative.

The difference in rise between the BLEU and NIST scores is due to the fact that NIST hardly gives any credit for correct word order, whereas BLEU gives too much credit for getting 3- and 4-grams right, overriding the contribution from unigrams (Zhang, Vogel, and Waibel, 2002).

The addition of ‘transfer’ rules, compared with the ‘no-rules’ condition leads to a relative rise of 18% in BLEU score and 9% in NIST score.

For some sentences, the generated translations degrade when adding the *transfer* rules. We need to further investigate these cases, and probably apply more strict conditions as to when the transfer rules fire.

In all, these results show that the addition of the *transfer* rules increases the system’s performance.

Figure 3: Best BLEU Scores

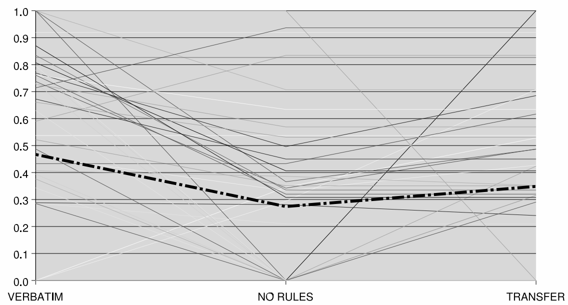
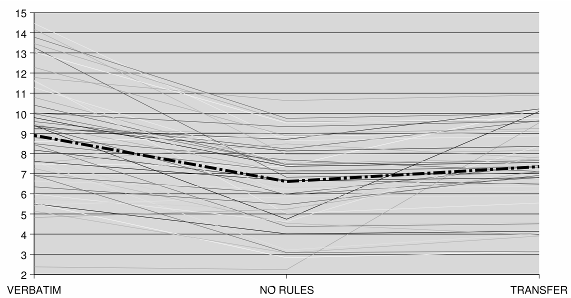


Figure 4: Best BLEU Scores



3.2.2 Best BLEU and NIST scores

The results for the best BLEU and NIST scores, as shown in table 5 and figures 3 and 4 are somewhat surprising.

Table 5: Best BLEU and NIST scores for the three conditions

	Verbatim	'no rules'	Transfer
BLEU	0.4669	0.2740	0.3486
NIST	8.9051	6.6103	7.3500

We find that, by performing word by word translations, there are translations among the (huge) list of translation alternatives, which yield better BLEU and NIST scores than the best scores in the other conditions.

As these *best* scores show the *potential* of the system under a certain condition, it shows that source language word order information should be included in the target language model, as the translation alternative with the highest BLEU/NIST score was generated in the 'verbatim' condition, where no target-language lookup was performed, and where each lexical alternative from the dictionary receives equal weight.

This is, of course, due to the fact that Dutch and English are closely related languages, and is not generalisable to translation pairs of more distant languages. Furthermore, note that we allow all permutations and have no chance to select a better one amongst those.

Currently we do not use any source language word order information, apart from the source language analysis.

While these results degrade when going from 'verbatim' to 'no-rules', and from 'verbatim' to 'transfer', there is still a relative improvement when going from 'no-rules' to 'transfer' of more than 21% in BLEU score and 10% in NIST score.

This shows that, when using corpus-based lexical selection and word reordering, the addition of *transfer* rules still leads to improvement.

The difference between *average* scores as described in section 3.2.1, and the result described in this section are a lot smaller for the 'no-rules' and 'transfer' conditions than for the 'verbatim' condition, as there are much less proposed top-weight translations in those conditions.

4 Conclusion and future directions

The most surprising findings from the experiment presented in this paper are the high scores for the highest scoring translation alternative in the condition in which we apply word for word translation. From these results, we can conclude that, for the language pair at hand (Dutch-English), the use of source-language word order information holds potential.

In our current system, lexical selection and word order information are derived from the target-language corpus using corpus matching weights as described in Dirix et al. (2007). In a future update of our system, we will separate the lexical selection from the word order information, so we can investigate the full potential of combining lexical selection, with source language word order information.

Apart from this, the addition of a few transfer rules to our 'no-rules' system clearly improves the output. Not only are the BLEU and NIST scores considerably higher than in our 'no-rules' system, the output becomes much more readable.

In the few cases where the performance decreases, we will investigate the reasons for this, which might result in applying more restrictions as to when the rules fire.

The addition of a few more rules can improve the system further: *do* insertion in the imperative mode and in yes/no questions is not yet implemented, and neither is the tense mapping towards continuous tenses, nor the handling of IPP.

The handling of coordinating conjunctions is still an issue, as our source language analysis does not have a good performance on detecting the scope of a coordination, which can lead to false translations. This is difficult to solve as we are only using *shallow* source language analysis.

We are currently updating our corpus, as we noticed that some mistakes have been made in the corpus preprocessing. We expect results to improve after this update has been done.

We will apply *subject detection* both on the sentence level and on the clause level. This will reduce the number of possible permutations, and hence reduce the number of output translations. It will also speed up the translation process, as less translation candidates will have to be investigated.

In all, we can conclude that the addition of a few rules to our data-driven system clearly improves performance. Not sticking to one paradigm, be it rule-based or corpus-based, but combining ideas coming from both worlds leads to better results.

References

Brants, T. (2000). TnT - a statistical part-of-speech tagger. In *Proceedings of the 6th applied NLP conference*, ANLP-2000.

Carl, M., Schmidt, P., and Schütz, J. (2005). Reversible Template-based Shake & Bake Generation. In *Proceedings of the 2nd Workshop on Example-based Machine Translation*. Phuket, Thailand

Dirix, P., Vandeghinste, V., and Schuurman, I. (2007). A new hybrid approach enabling MT for languages with little resources. In *Proceedings of Computational Linguistics in the Netherlands 2005*.

Doddington, G. (2002). Automatic Evaluation of Machine Translation Quality using N-gram Co-occurrence Statistics. In *Proceedings of the second Human Language Technologies Conference* (HLT-02). San Diego, pp. 128-132.

Haeseryn, W., Romijn K., Geerts, G., De Rooij, J., and Van den Toorn, M.C. (1997). *Algemene Nederlandse Spraakkunst*, Martinus Nijhoff, Groningen.

Huddleston, R., and Pullum, G. (2002). *The Cambridge Grammar of the English Language*. Cambridge University Press.

Papineni, K., Roulos, S., Ward, T., and Zhu, W. (2001). BLEU: a method for automatic evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Asosciation for Computational Linguistics*, ACL, Philadelphia.

Vandeghinste, V. (2003). Rule-based nonrecursive chunking for Dutch. *Computational Linguistics in the Netherlands*. Abstracts. Antwerp.

Vandeghinste, V. (2005). *ShaRPa2.0. User Manual*. Centre for Computational Linguistics, Leuven.

Vandeghinste, V., Schuurman, I., Carl, M., Markantonatou, S., Badia, T. (2006). METIS-II: Machine Translation for Low Resource Languages. In *Proceedings of the 5th International Conference on Language Resources and Evaluation* (LREC).

Van Eynde, F. (2005). *Part of Speech Tagging en Lemmatisering van het D-CoI corpus*. Annotation Protocol. Centrum voor Computerlinguïstiek, Leuven.

Yu, S. (1993). Automatic Evaluation of Output Quality for Machine Translation Systems, *Machine Translation*, vol. 8, pp. 117-126. Kluwer

Zhang, Y., Vogel, S., and Waibel., A. (2004) Interpreting BLEU/NIST Scores: How Much Improvement Do We Need to Have a Better System? In *Proceedings of the 4th International Conference on Language Resources and Evaluation* (LREC). pp. 2051-2054.