View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by St Andrews Research Repository

THE MAT SAT ON THE CAT: INVESTIGATING STRUCTURE IN THE EVALUATION OF ORDER IN MACHINE TRANSLATION

Martin McCaffery

**A Thesis Submitted for the Degree of PhD
at the
University of St Andrews**

2017

**Full metadata for this item is available in
St Andrews Research Repository
at:
http://research-repository.st-andrews.ac.uk/**

**Please use this identifier to cite or link to this item:
http://hdl.handle.net/10023/12080**

This item is protected by original copyright

**This item is licensed under a
Creative Commons Licence**

The mat sat on the cat: Investigating structure in the evaluation of order in machine translation

by

Martin McCaffery

University of
St Andrews

FOUNDED
1413

This thesis is submitted to the

UNIVERSITY OF ST ANDREWS

in partial fulfilment for the degree of

DOCTOR OF PHILOSOPHY

submitted on

23rd June 2017

Copyright © 2017 by Martin McCaffery

Abstract

We present a multifaceted investigation into the relevance of word order in machine translation. We introduce two tools, DTED and DERP, each using dependency structure to detect differences between the structures of machine-produced translations and human-produced references.

DTED applies the principle of Tree Edit Distance to calculate edit operations required to convert one structure into another. Four variants of DTED have been produced, differing in the importance they place on words which match between the two sentences. DERP represents a more detailed procedure, making use of the dependency relations between words when evaluating the disparities between paths connecting matching nodes.

In order to empirically evaluate DTED and DERP, and as a standalone contribution, we have produced WOJ-DB, a database of human judgments. Containing scores relating to translation adequacy and more specifically to word order quality, this is intended to support investigations into a wide range of translation phenomena.

We report an internal evaluation of the information in WOJ-DB, then use it to evaluate variants of DTED and DERP, both to determine their relative merit and their strength relative to third-party baselines. We present our conclusions about the importance of structure to the tools and their relevance to word order specifically, then propose further related avenues of research suggested or enabled by our work.

Candidate's Declaration

I, Martin McCaffery, hereby certify that this thesis, which is approximately $57,000$ words in length, has been written by me, and that it is the record of work carried out by me, or principally by myself in collaboration with others as acknowledged, and that it has not been submitted in any previous application for a higher degree.

I was admitted as a research student in March 2013, and as a candidate for the degree of Doctor of Philosophy in June 2017; the higher study for which this is a record was carried out in the University of St Andrews between 2013 and 2017.

Date .... Signature of Candidate ....

Supervisor's Declaration

I hereby certify that the candidate has fulfilled the conditions of the Resolution and Regulations appropriate for the degree of Doctor of Philosophy in the University of St Andrews and that the candidate is qualified to submit this thesis in application for that degree.

Date .... Signature of Supervisor ....

Permission for publication

In submitting this thesis to the University of St Andrews I understand that I am giving permission for it to be made available for use in accordance with the regulations of the University Library for the time being in force, subject to any copyright vested in the work not being affected thereby. I also understand that the title and the abstract will be published, and that a copy of the work may be made and supplied to any bona fide library or research worker, that my thesis will be electronically accessible for personal or research use unless exempt by award of an embargo as requested below, and that the library has the right to migrate my thesis into new electronic forms as required to ensure continued access to the thesis. I have obtained any third-party copyright permissions that may be required in order to allow such access and migration, or have requested the appropriate embargo below.

The following is an agreed request by candidate and supervisor regarding the publication of this thesis:

No embargo on any electronic nor print copy.

Date .... Signature of Candidate ....

Signature of Supervisor ....

This thesis is dedicated to my father, Peter McCaffery,
who gave me an unquenchable desire to learn.

ACKNOWLEDGEMENTS

One name on the cover page doesn't quite represent this thesis. I may have done the actual researching and writing, but without countless discussions with my supervisor, Mark-Jan Nederhof, this could never have got to submission. Tom Kelsey, my second supervisor and frequent victim (or cause) of research-related distractions, provided a valuable extra perspective on the work.

If I'm mentioning perspectives, distractions, interesting discussions and idea-bouncing, it's worth noting that all of that was provided by the rest of the School too. Ruth Letham, Fearn Bishop, Ruth Hoffmann, Adam Barwell, Stuart Norcross, Ian Gent, and plenty more – including, last but by far not least, my officemates and seemingly adopted siblings Shyam Reyal and Vinodh Rajan. You all helped more than you know to make the last few years thoroughly interesting, useful and enjoyable.

And while the actual research is a key part of the Ph.D. process, it's also pretty important to be able to communicate the work to non-academics. For that, I'd like to thank Malinda Kathleen Reese for creating the wonder that is Google Translate Sings!

I should stop naming names: there are too many people. Except, it's absolutely necessary to give an enormous thank-you to Cat Doyle and Laura Price, my academic daughters and, um, mothers?: helpful, constant, understanding, supportive... you know how much you matter, but it's worth mentioning anyway!

Life in St Andrews can be busy sometimes, and many of the most important moments of the last few years have been in social spheres. Thanks to members of the Real Ale Society (and the James Robb Avenue group) for the ability to drink away the (occasional) sorrows of writing and discuss all things (un)related to research. And thanks to my local blues communities for helping me be grounded and relaxed through the recent years.

For more ever-present and/or personal support, there are a few more names to mention. The Quakers, of course: quite apart from giving me one of the best places to live in town, both individual Friends and the Quaker outlook on and practice of life have been constantly with me. Also, Fiona Howe, Hannah Jones and other Edinburgh friends: it's been a pleasure to borrow your floors, evenings and ears. Closer to home, the Celtic Society dancers including Stef Eminger and Hannah Mace have pushed me to new heights of dance and of understanding.

Thanks also to everyone I've met through Taizé, whether long-time friends or passing encounters, for keeping me sane (or just pleasantly insane?) during my various escapades throughout Europe and even Britain over the last few years.

And finally, my actual family. My brother John has been an impressive example to follow, and my mother has provided more insights, comments and just support than I can say. As of course my father, an example of a proper academic, always keen to gain and pass on knowledge, has been an inspiration for much more than the duration of this PhD.

Contents

List of Figures

List of Tables

CHAPTER ONE

INTRODUCTION

1.1 Introduction

In this thesis, we explore the evaluation of automatically produced translations. Many approaches exist both for producing such translations and for assessing their quality. While most evaluation tools inspect immediately-visible syntactic features such as words, a small number rely on grammatical structures for their processing. Additionally, most existing tools assess the overall quality of sentences, although more recently some have focused on specific aspects of the sentences such as their word ordering.

We introduce two tools, DTED and DERP, which combine these more novel approaches: both evaluate machine translation using grammatical structure, with a view to assessing the quality of word ordering in particular. To judge their accuracy we have produced a bespoke dataset, WOJ-DB, which contains 'gold standard' human assessments of translations' word ordering and overall quality: these are later compared both with our own metrics and with various third-party packages.

1.2 Thesis overview

This section will provide a high-level summary of the context, contributions, goals and results of our investigations. It is primarily intended for those who are familiar with the area: should the reader require a more in-depth introduction to the domain, they are directed to Chapter 3.

Automatic assessment – The field of machine translation has experienced enormous growth and innovation over the last few decades. One such innovation is the development of automatic tools for the evaluation of the quality of translated sentences, which has in many ways become key to the development process of new machine translation systems.

Order metrics – As the variety of both translation systems and evaluation metrics widens, a demand has arisen for tools which measure more fine-grained information than simple holistic translation quality. While a number of tools attempt to categorise errors to give such deeper knowledge, the importance of one specific error type – word ordering – has become clear, meriting bespoke tools for its evaluation alone.

Structural metrics – A number of such tools exist, both adaptations from other domains and purpose-built metrics using common evaluation techniques. One technique which is becoming more common is the use of structural features rather than purely surface-level syntax, yet to our knowledge only one tool exists which uses this rich resource for the evaluation of ordering.

Research questions – Our work represents an effort to rectify the general lack of such structure-based ordering metrics. We have produced two separate metrics, DTED (Chapter 4) and DERP (Chapter 5), both based on dependency parse structures. By using and evaluating these tools, we intend to find responses to three distinct questions:

1. High performance – Can we improve on the current state of the art of word order evaluation?

2. Relevance of structure – Does structure aid the evaluation of word order in machine translation?

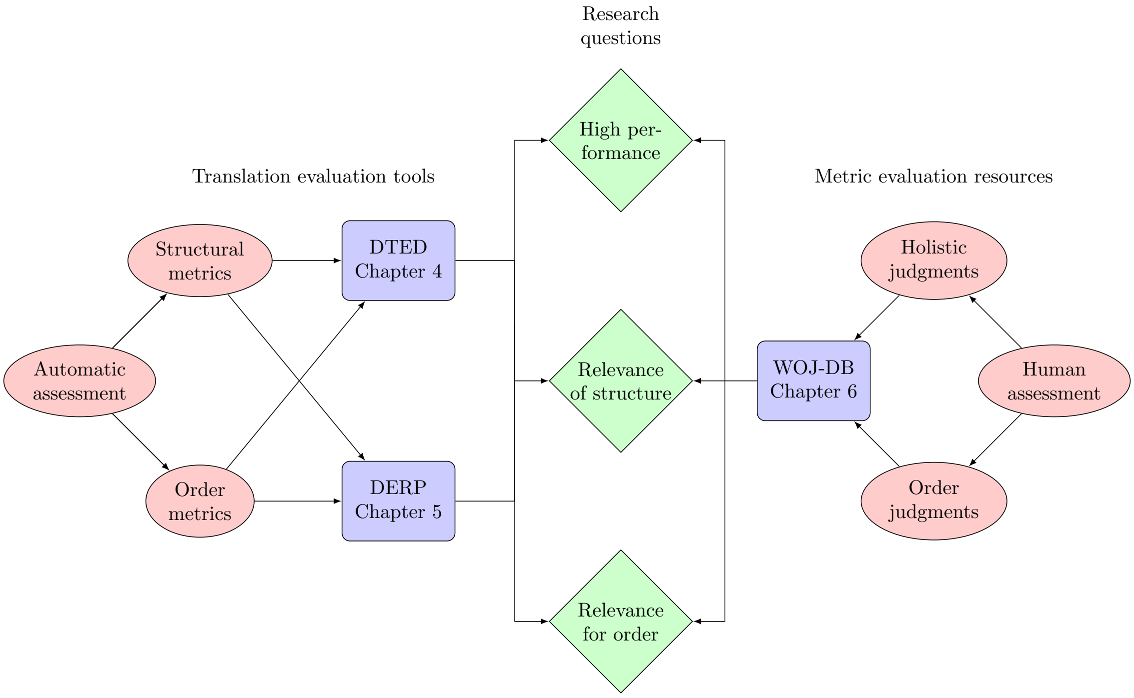


Figure 1.1: Summary of thesis contents. Rectangles indicate significant contributions, diamonds refer to our three research questions, while ovals provide context.

3. Relevance for order – Does dependency structure permit word order evaluation, or does it lend itself more to holistic judgment?

DTED – The approach of DTED, the simpler of our two tools, is based on the principle of Tree Edit Distance. Something of an adaptation of Levenshtein distances to the domain of tree structures, these consist of a count of simple deletion, insertion and substitution operations required to convert one tree to another.

Through DTED, we apply this principle to the comparison of parse trees generated from a machine-produced hypothesis translation and a human-produced reference. We produce normalised scores which can be easily compared between sentences and translation systems. Through variations in the treatment of words which match between the two sentences (‘aligned’ words), we provide four variations on this fundamental concept.

DERP – Our second tool, DERP, is intended to leverage more information than that used by DTED: namely the labels indicating functional relationships between words in a dependency parse tree. It does this by comparing the paths between pairs of nodes in one tree to pairs of aligned nodes in the other, evaluating differences using traditional Levenshtein distances.

Enough paths contribute to the final DERP score such that every aligned node is compared, either directly or indirectly, to every other. The exact choice of paths to include is determined through Kruskal’s algorithm for the minimum spanning tree. The Levenshtein costs for such paths are then normalised, as before, to produce an easily comparable score in the range [0,1].

DTED & DERP: relevance of structure – In order to unambiguously investigate the rôle played by structure in the performance of our tools, we have produced variants of each which ignore all structural information, applying their algorithms on ‘flat’ sentences. We also abstract away from the relevance of third-party tools – taggers, parsers and alignment generators – through the easy replacement of each of these components.

WOJ-DB **–** The evaluation of tools intended to measure order specifically is no easy task. While numerous datasets, and even an annual conference, provide human judgments on holistic quality which are used for metric evaluation, error-specific information is much rarer.

We have thus generated our own dataset, WOJ-DB (Chapter 6). We have done this through a survey, asking local people of various backgrounds to rate the overall quality and the ordering quality of a number of translations. Participants were provided with reference and hypothesis translations in English only, with sentences sourced from the annual Workshop on Machine Translation.

WOJ-DB: relevance of structure **–** While participants were asked to rate the quality of a large number of these sentences directly, another form of ‘translation’ was included in the surveys. As the mistakes in real translated sentences are numerous and often interact in complex ways, it can be difficult to isolate the relevance of specific types of inaccuracy, even to a question as specific as word order quality. We have thus automatically permuted correct sentences, introducing simple errors such as word swaps, to allow for a more precise analysis and understanding.

Results **–** We have performed a number of analyses of the data gathered for WOJ-DB (Section 7.2), and hope to make the information available for future researchers in the area of word order in machine translation. More immediately, however, we have used it to evaluate the tools we have produced: DTED and DERP (Section 7.3).

Results: high performance **–** In terms of their ability to predict human judgments of word order, we have obtained encouraging results: one variant of DTED outperforms all baselines we compare against: Kendall’s τ, BLEU and Meteor. Most other variants of our tools perform at levels comparable with those baselines, although interestingly our more complex tool, DERP, performs strictly less well than one.

Results: relevance of structure **–** When considering structure, our results are potentially surprising. While DERP and one variant of DTED achieve – as predicted – a higher level of accuracy when provided with structural information, the same is not true for the three higher-performing variants of DTED. For these three tools, omission of information relating to the syntactic structure of the sentence barely affects their success.

Results: relevance for order **–** The results relating to our third research question, the relative ability of structure to aid in predicting holistic and ordering quality, are striking. Almost all variants of both our tools score almost exactly equally highly in their ability to predict each of the two scores provided by participants to WOJ-DB.

Conclusions **–** All three of these results suggest that the combination of structure and word order evaluation is an interesting one. Through the success of our tools we have shown it has merit, while the more mixed results we have received for our latter two research questions demand more investigation to be fully understood.

Given these results, we encourage the translation community to consider our and similar techniques in future metrics, and to further our own investigations through the adaptation of our tools to other languages, datasets and more.

Chapter Two

Machine Translation Evaluation: Existing Work

2.1 Natural Language Processing

Given the pre-eminence of language in so many aspects of our everyday lives, it is perhaps unsurprising that the computational understanding and treatment of natural language has been actively investigated since long before the advent of electronics [Bhate and Kak, 1991]. Indeed, the question of translation from one language to another has in particular been of practical relevance for thousands of years [Shieber, 2007].

More recently, the invention of the computer has provided hitherto unheard-of resources for deepening our knowledge of language, and the field of computational Natural Language Processing (NLP) has been active since not long after computers became available [Hays, 1962; Weaver, 1955]. Nowadays, it has several disparate yet interlinked sub-fields.

2.1.1 Grammar formalisms

One of these subfields with the most direct effect on the level and types of processing which can be done is that of designing grammar formalisms: ways of representing language according to computational structures and other features.

Of these, two of the simpler options in widespread use are Regular Expressions and Finite State Automata (FSAs) [Roche and Schabes, 1997]. Computationally equivalent to each other, these are convenient as programming tools yet severely limited in their ability to model the real-world complexities of language.

In practice, we often require more modeling power than these formalisms can provide. One of the most common grammar formalisms in use today, Context-Free Grammars (CFGs) [Hopcroft *et al.*, 2001], applies to words and phrases rather than the simple characters involved in FSAs. CFGs are able to model nearly all phenomena in natural language, representing a well-known and well-understood class of grammars [Kallmeyer, 2010]: they lend themselves to situations where complex features of language may occur but parsing speed and grammar size are of high importance.

CFGs encode relationships between elements of a sentence in simple nestable rules, defining each phenomenon – for example, a noun phrase – in terms of its constituent parts: a proper noun, perhaps, or an adjective and plural noun. These example rules could be written as $[NP \rightarrow NNP]$ and $[NP \rightarrow JJ\ NNS]$ respectively.

A number of formalisms opt to represent linguistic phenomena in ways other than through categorisation of words or phrases. One well-known example of this is the class of Dependency Grammars [Gaifman, 1965], which, though computationally equivalent to CFGs [Rambow, 2010], use a very different approach. The tree structures generated by dependency parses begin with a key verb of the sentence and recursively link this to its direct 'dependents' – such as a subject or object – with connections which may be either unlabelled or annotated with the nature of the relationship. We discuss dependency structures further in Section 3.4.3.

Context-Free and Dependency Grammars, while powerful, are far from the most expressive formalisms we know. Tree Adjoining Grammars (TAGs) [Joshi *et al.*, 1975] are able to model many of the most unusual features of natural language, while Linear Context-Free Rewriting Systems (LCFRSs) [Vijay-Shanker *et al.*, 1988], though very heavyweight, demonstrate even more ability to model complex edge cases [Kallmeyer, 2013].

Other grammar formalisms may approach the question of language representation in more ways again. While CFGs, TAGs and LCFRSs focus on relationships between words according to their positions in sentences, Hyperedge Replacement Grammars [Chiang *et al.*, 2013] and Combinatorial Categorical Grammars [Steedman, 2000] are intended to abstract away from the words themselves and instead encode more generic, semantic features.

2.1.2 Grammar inference

While important, choosing an appropriate formalism is only the first step in language processing: before a formalism may be used, it must be instantiated to relate to a specific language. While in the past this has been achieved through hand-crafted rules [Klein and Simmons, 1963], more recent advances in computational power have allowed grammars to be stochastically 'trained' or 'inferred' from an appropriate corpus [Nadkarni *et al.*, 2011].

Since the inception of this statistical approach, many different methods have been explored. These can vary widely based on the grammar formalism used [Graehl *et al.*, 2004; Kwiatkowski *et al.*, 2010], but most have a number of procedures in common.

First, a large corpus of existing text is examined: this is usually made up of sentences which are already represented as trees (a 'treebank'), in which the words and other elements may or may not be annotated with more relevant information such as parts of speech [Nederhof and Satta, 2008; Klein and Manning, 2003b]. The features common between different trees can be extracted and directly leveraged to produce grammars, with the frequencies of such features then used to estimate the probabilities of their occurring in hitherto-unseen sentences [Manning and Schütze, 1999].

While treebank-based 'supervised' learning is very common, it requires potentially enormous numbers of sentences prepared in many cases by human experts. 'Unsupervised' learning is instead based on raw sentences with little additional information [Clark, 2003; Klein and Manning, 2004], while 'weakly supervised' techniques attempt to use only minimal information beyond a relatively small core of such sentences [Druck *et al.*, 2009]. These approaches involve relatively arbitrary initial rules being refined through iterative procedures such as EM [Dempster *et al.*, 1977] as applied in the Inside-Outside algorithm [Lari and Young, 1990].

Grammars produced through supervised or unsupervised learning techniques may be further refined using various techniques, e.g. coarse-to-fine retuning [Petrov *et al.*, 2006], until the resulting grammar is deemed to fit the training data as well as reasonably possible without overfitting compromising its ability to predict unseen data. This may or may not result in a provably optimal representation of the corpus [Nederhof and Satta, 2004].

Depending on the language being investigated, the size of the training corpus can vary dramatically, as can the amount of information offered for each word or tree element. For English, several corpora exist [Souter and Atwell, 1994], most of which follow the well-known phrase-structure format [Francis, 1964]. The most popular of these is the Penn

Treebank [Marcus *et al.*, 1993], whose data has also been adapted for a number of more niche projects [Hockenmaier and Steedman, 2007; Weischedel *et al.*, 2011].

2.1.3 Parsing techniques

Once an appropriate grammar has been extracted in its entirety, it can be used to parse individual sentences. Given the sizes of grammars which represent entire languages, even after processing through the techniques mentioned above, this is difficult to do with both speed and accuracy. A large amount of work has been done in attempting to increase one or both of these properties [Nederhof and Satta, 2006; Cer *et al.*, 2010].

While individual parsing techniques can vary hugely in their approaches, several general trends have gained widespread popularity. One which applies to nearly all parsers is that they be probabilistic in nature, in order to capture the inherent ambiguity common in real-world sentences [Manning and Schütze, 1999]. They then generally fall into one of two key categories: top-down and bottom-up.

In the former case, a top-level symbol (such as the catch-all 'start symbol' for a CFG) is expanded incrementally, working to determine the appropriate children for each node [Roark, 2001]. The latter, on the other hand, begins with bottom-level terminal and preterminal symbols, combining them recursively until the top-level symbol is reached.

One of the most widely referenced parsing techniques is the CYK algorithm [Younger, 1967], which operates on CFGs in Chomsky Normal Form (CNF) [Sipser, 1996]. It has a straightforward yet relatively efficient bottom-up approach: after matching every terminal symbol first to nonterminals of length 1, it iteratively groups these nonterminals according to rules in which they occur, producing ever-larger, ever-higher-level groups.

While bottom-up parsers are common in NLP, the top-down approach is generally used only within deterministic contexts such as that of programming language parsing. Partial exceptions to this exist, however, such as the Earley parser [Earley, 1970] which contains prediction elements based on a top-down model.

One of the most common techniques in the field of programming languages is shift-reduce parsing [Schabes, 1991], often put into practice as LR parsing [Knuth, 1965]. While these are not directly applicable to the class of natural languages due to their ambiguity and consequent nondeterminism, they have been generalised to apply to that more complex domain [Tomita, 1991].

The way any shift-reduce parser works is to progress through the input, token by token, deciding at each step whether to perform a Shift action – adding the current token to a stack and continuing – or a Reduce. This action replaces one or more tokens on the stack with a higher-level aggregate nonterminal.

Countless other parsing procedures exist with various differences and improvements from these and other paradigms. Many expand their applications to formalisms other than CFGs [Chiang *et al.*, 2013; Gómez-Rodríguez *et al.*, 2011; Clark *et al.*, 2002], or provide other methods of speeding up their processing [Klein and Manning, 2003a].

It should be noted that parsers generally work on input which has been processed in some ways, to allow certain abstractions to be made by the algorithm. One fundamental requirement of most parsers, for example, is a 'tokenisation' step to separate the input into manageable atomic units. This can be much more difficult than merely splitting a sentence by whitespace characters, due to edge cases related to punctuation, lack of spaces between words in some languages, etc. [Grefenstette and Tapanainen, 1994].

Most NLP parsers also include other normalisation steps, which ensure consistency of capitalisation, encoding and such: these features should be consistent both within the text and with the dataset used for grammar inference [Clark, 2003]. A third type of common preprocessing step is part-of-speech tagging [Voutilainen, 2003]: detecting from position and other features the function of a word within a sentence.

Given the wide range of parsing approaches, requirements and datasets, it can be daunting to decide which to use for any of the wide variety of situations where parsed sentences are required. Furthermore, it may be simply unfeasible to produce a bespoke parser for each and every NLP project.

Happily, a number of open-source parsers and other resources have been produced and are freely available. These include the Stanford CoreNLP project [Manning *et al.*, 2014], Apache OpenNLP [The Apache Software Foundation, 2011], and the Python Natural Language Toolkit (NLTK) [Bird, 2006]. Each of these provides a range of tools, although other projects simply focus on addressing one aspect of the parsing process [Nivre, 2003; Och and Ney, 2003].

2.1.4 Machine translation

While there are a large number of problem types which can be addressed through the use of NLP techniques, our project is placed specifically within the subdomain of Machine Translation (MT). The fundamental issues of MT are in many cases inherited from NLP [Lopez, 2008], although their priorities and practical details are often dramatically different from those related to the more general research field [Tsujii, 1986].

As with many areas of NLP, MT has been approached in numerous ways over the years. The earliest of these was the hand-crafting of rule-based techniques [Johnson *et al.*, 1985], with links between languages being defined by human experts. This approach has achieved moderate success and has not entirely disappeared [Scott and Barreiro, 2009; Forcada *et al.*, 2011], although it has been largely superseded by techniques whose exact behaviour is based on multilingual datasets similar to monolingual corpora.

In the multilingual domain, such datasets are known as ‘parallel corpora’, ‘bitexts’ or ‘Hansards’ [Brown *et al.*, 1990]. They contain large numbers of pairs of sentences for which each is a translation of the other, either because one was produced by a human translator who was shown the other, or because both were the result of translators given a ‘source’ sentence in a third language. In practice, some parallel corpora include sentences following this format in more than two languages [Koehn, 2005].

Generally, prior to publication such corpora are curated manually to ensure that the sentences are truly related: in practice, edge cases such as a single sentence in one language translated into two in another can cause confusion when training translation systems. Automatically detecting sentence pairings and ensuring strict (usually one-to-one) matches is far from trivial, and is the goal of the field of sentence alignment [Brown *et al.*, 1991; Fung and Church, 1994].

Several multilingual parallel corpora exist and are available for use in translation systems. These include Europarl [Koehn, 2005], a collection made from transcriptions of the proceedings of the European Parliament over the course of several years. The copyright freedom and abundance of multidirectional translation have made governmental records a popular source of high-quality translations, with the JRC-Acquis corpus also based on public data [Steinberger *et al.*, 2006]. Other resources also exist which contain data for just two languages [Čmejrek *et al.*, 2004; Chen and Nie, 2000].

Among the first major attempts to make use of parallel corpora for machine translation were the IBM models [Brown *et al.*, 1993]. These model translation as a series of separate sub-tasks, such as word choice and ordering. Sub-tasks make use of separate tables of parameters, each trained on a bitext using the EM algorithm. Successive models introduce more complexity and relate the translation to the source sentence in more ways.

The IBM models have over the years been adapted and improved in a number of ways, most notably by the Yamada-Knight system [Yamada and Knight, 2001]. This is strongly based on the original models, but uses a parse tree as an input rather than a flat sentence, and is able to perform much more powerful reordering, insertion and translation steps.

One notable feature of the IBM models is that they operate at the level of words: these units can be translated, moved or otherwise manipulated independently of each other. This provides flexibility, but also limits the systems. The introduction of phrase-based models, which involve translating multiple words together before combining the results into a full sentence, was thus a major innovation [Zens and Bender, 2005; Yang and Kirchhoff, 2006]. A publically available phrase-based system, Moses, has been developed by researchers primarily based in Edinburgh [Koehn *et al.*, 2007].

While word- and phrase-level systems are well-known, they are not the only methods of performing translation. In recent years, for example, the use of neural networks has achieved remarkable success [Bahdanau *et al.*, 2015; Jean *et al.*, 2015; Sennrich and Haddow, 2016].

However, more common than these are systems which use statistical or 'example-based' solutions in ways similar to monolingual parsing [Sumita and Iida, 1991; Somers, 1999; Koehn, 2010]. The fundamental elements required for such translation remain as described in previous sections, beginning with a grammar formalism and training corpus being selected or devised. Both of these can be as flexible as in the monolingual case, but must be adapted to treat multiple languages.

While in the monolingual domain there are a multitude of different grammar types of varying expressivity and complexity, the same is true of the multilingual case. Many such formalisms [Nederhof and Satta, 2011; Büchse *et al.*, 2011; Nesson *et al.*, 2006] are simply 'synchronous' forms of single-language techniques [Chiang and Knight, 2006].

In a synchronous grammar, instead of rules simply relating a phenomenon to its constituent parts, a pair of such rules from each language are linked. For example, a CFG rule $[\mathit{NP} \rightarrow \mathit{JJ}\ \mathit{NNS}]$, indicating that a noun-phrase may be composed of an adjective followed by a plural noun, might become $[\mathit{NP} \rightarrow \mathit{JJ}\,\mathit{NNS}, \mathit{NP} \rightarrow \mathit{NNS}\,\mathit{JJ}]$ in an SCFG. This would indicate that the relative orderings of the two components JJ and NNS are swapped between the languages. Note that the process of generating a synchronous grammar is closely related to the monolingual equivalent [DeNeefe and Knight, 2009].

Many of the different approaches from monolingual parsing can also be found in the context of translation, sometimes through the use of synchronous grammars. For example, the move from part-of-speech-based parsing to dependency structures has been made through a variety of systems [Ding and Palmer, 2005; Galley and Manning, 2009; Gimpel and Smith, 2011].

2.2 Translation evaluation

While research into different methods, formalisms and styles of translation has received significant attention for several decades now, the area of evaluation of the resulting systems is more nascent. When evaluating machine translation, two broad aspects are often considered. ‘Adequacy’ refers to the extent to which the meaning of the original source sentence is clear within the translated hypothesis, while ‘fluency’ considers only whether that hypothesis represents grammatically correct and idiomatic use of the target language [Koehn, 2010, p. 218].

Both of these measures, as well as any combined measure of the overall ‘goodness’ of a translation, are obviously highly subjective, with no guarantee that even trained judges will agree on the appropriate score for a given sentence. Despite this, they are arguably the only true measures of quality. Partially as a result of that argument, before the last two decades subjective human ratings were the primary method of translation evaluation [Al-Onaizan *et al.*, 1999; Papineni *et al.*, 2002].

The procedure of generating such human evaluations can be streamlined using helpful tools [Nießen *et al.*, 2000], and has the undeniable advantage that the human translator’s complete understanding would be difficult to fool by any specially-engineered techniques. However, a powerful issue with human judgments is that their generation requires vast time investments, rendering them prohibitively impractical for frequent use – such as that needed for the iterative development and training of a new system [Koehn, 2010, p. 222].

2.2.1 Precision & recall metrics

To address the need for some sort of efficient technique for judging translations, various automatic systems have been designed. The goals of these are to predict the judgements a hypothetical reasonable human would make of the quality of sentences, and indeed the evaluation of such tools generally involves comparing the scores they provide with similar scores provided by human evaluators.

One of the earliest, and arguably the most popular, automatic evaluation tool is the BLEU metric [Papineni *et al.*, 2002], which counts the phrases in the system translation which have exact matches in the reference(s). Phrases are considered in groups of n adjacent words (‘n-grams’), from which a score is produced using a version of the information-retrieval concept of precision [Ting, 2010].

Precision is a common technique for evaluating the success of a search query. It is calculated as a ratio between the number of returned items which are considered ‘relevant’ – in BLEU, the number of n-grams in a machine-produced hypothesis translation which also occur in a human-produced reference – compared with the total number of returned items (n-grams) in the hypothesis.

To avoid giving unreasonably high scores to short sentences, for which the total number of n-grams will be low, BLEU multiplies the precision score by an exponential ‘brevity penalty’ based on the difference in lengths of the two sentences. It also allows for multiple different lengths of n-gram to be calculated and combined, with the best-performing variants combining results for n from 1 to 4. Finally, it also permits multiple reference translations to be used in an attempt to allow for legitimate variations in phrasing. An example of BLEU being applied to a sentence pair can be found in Section 3.5.3.

Since the emergence of BLEU, many other tools have been devised which similarly take advantage of the concept of precision. Many couple it with its cousin recall, sometimes aggregating the two using their harmonic mean in the F-score or F-measure [van Rijsbergen, 1979]. Recall is the ratio between the number of relevant hypothesis n-grams to the total number of n-grams in the reference translation(s) instead of the hypothesis. Melamed *et al.* [2003] investigated the use of such formulae on their own as evaluation mechanisms.

While precision and recall are often calculated on words or groups thereof, this is far from an absolute trend. Popović and Ney [2009] tried applying all four concepts (precision, recall, F-score and BLEU) to part-of-speech tags produced by tagging tools, with partial success. rgbF [Popović, 2012b] more generically allows mixed types of word-level n-gram matches, such as part-of-speech tags or word stems (base forms). More recently still, CHRF [Popović, 2015] applies them at the level of character n-grams.

The techniques used to determine which words 'match' between two sentences can vary dramatically within recall- and precision-based metrics. Some of the criticisms of BLEU observe that it simply compares surface forms of words: thus, 'begin' will match only 'begin' and not 'began', 'begun', etc. This lack of linguistic knowledge works in BLEU's favour in some ways, as it allows the tool to be applied to translations in any language [Chen and Kuhn, 2011], but it also imposes severe practical limits.

Such limits can be lifted in a number of ways, with words being 'aligned' together using more features than just their exact syntactic form. This is explored in the active field of word alignment [Melamed, 2000; Vilar *et al.*, 2006a], with numerous large projects existing whose primary purpose is to produce appropriate mappings of words between sentences [Och and Ney, 2003; Liang *et al.*, 2006]. Some of these are discussed in more detail in Section 3.4.5.

Word alignment can be performed through a number of methods. These can involve statistical observations of co-occurring words in large bitexts [Och and Ney, 2003], or may rely on major resources such as the word-linking project WordNet [Miller, 1995]. Many bespoke systems used by individual metrics employ much simpler techniques such as 'stemming' to produce generic forms of words [Lavie *et al.*, 2004]. These techniques are usually highly language-dependent, although it has been shown that moderate performance can be achieved through entirely language-agnostic measures [Popović *et al.*, 2015].

One popular tool which relies on these techniques is Meteor [Banerjee and Lavie, 2005; Lavie and Agarwal, 2007], which addresses the problem of alignment through the cascaded use of three alignment techniques. Using exact matching, word stem comparisons and WordNet synonyms [Miller, 1995], Meteor selects the largest possible set of aligned words to be used for its F-measure calculation. This results in a system which is language-dependent and heavier-duty than BLEU, yet considerably more robust. It also includes a secondary step to account for word ordering; this is discussed in more detail in Section 2.3.1.

Like many other metrics, Meteor allows for multiple reference translations through the simple technique of calculating its score using each, then using the most positive as its final result. It also allows for empirical 'tuning' of numerous parameters, such as the exact brevity penalty and the relative importance of precision and recall in the F-measure, depending on the language being used. More detail about all aspects of its algorithm, including an example, can be found in Section 3.5.4.

Another approach for refinement of BLEU is to apply weightings to n-grams, prioritising those which are considered more important – for example, less common ones – over others. While BLEU has itself been extended in this way [Babych and Hartley, 2004], the NIST metric [Doddington, 2002] incorporates this while also using a somewhat different calculation when producing its brevity penalty. While NIST has not become as popular as BLEU, it is nonetheless a common baseline when evaluating new metrics.

2.2.2 Error rate metrics

While n-gram matching has been proved to be a powerful method for evaluating machine translation, it is far from the only possible technique. A number of others exist, including evaluation through neural networks [Guzmán *et al.*, 2017]; we will examine two such alternative approaches, beginning with that of error rate calculations. Contrary to the techniques mentioned thus far, which focus on observing the similarity between elements in sentences, approaches centred on error rates instead phrase the question of evaluation in terms of conversion from one sentence to another.

This is generally done through a combination of simple operations. The most well-known technique of this form, Levenshtein distances [Levenshtein, 1965], was designed for generic ordered and labelled sequences, though in translation these are generally words in a sentence. It allows elements in one sequence to be deleted, inserted or substituted (renamed or replaced) to match those in another. Such operations are assigned costs (traditionally they are equally weighted) and the minimum total edit cost is produced.

For example, the sentence *the athlete jumped high* can be edited to match *the young athlete leapt* using a total of three operations, in one of two ways. Either *young* must be inserted between *the* and *jumped*, *high* must be deleted, and *jumped* must be replaced with *leapt*; or the words *athlete*, *jumped* and *high* could simply be replaced by *young*, *athlete* and *leapt* respectively. No sequence of two or fewer operations can equate the two sentences.

Within translation evaluation, Word Error Rate (WER) [Wagner and Fischer, 1974; Marzal and Vidal, 1995] represents the use of traditional Levenshtein distances applied to words, with the maximum possible number of edits used as a normalisation factor to produce a score in the range [0,1] which can be meaningfully compared between sentences. Position-Independent Error Rate (PER) [Tillmann *et al.*, 1997] ignores the ordering of words by instead reflecting primarily the number of occurrences of identical words in both hypothesis and reference.

Slightly more recently, error rate metrics have been produced which allow block movement: shifting of groups of words together for a cost lower than that which would be incurred using WER. Due to the inherent computational complexity of calculating such an augmented error count, CDER [Leusch *et al.*, 2006] relaxes an internal constraint for the sake of a more efficient algorithm, while Translation Error Rate (TER) and Human-targeted Translation Error Rate (HTER) [Snover *et al.*, 2006] use a greedy algorithm to find an approximate solution. This last includes human understanding in the judgment process, aiming to find the minimum number of edits a human must make before the two sentences' *meanings* are equivalent. TER has since been augmented to allow paraphrase matching and synonymy [Snover *et al.*, 2009].

Various other alterations have been made to the principle of using error rates for MT evaluation. These range from the use of multiple reference translations at once [Akiba *et al.*, 2001], through separation of deletion and insertion errors [Popović and Ney, 2007], to the incorporation of probabilistic finite state machines (pFSMs) to introduce nondeterminism; these are used to train an editing model to match human scores in the SPEDE metric [Wang and Manning, 2012].

2.2.3 Structural evaluation

While we can evaluate sentences based simply on the words in them – or features of those words, such as stems or parts of speech – more information can be gleaned about syntactic relationships between words by using parsing techniques, as discussed in Section 2.1.3.

For example, dependency relations provide a wealth of information about the purpose or broad role of the words they link. This is leveraged in MAXSIM [Chan *et al.*, 2008], where dependency labels are compared alongside WordNet synonyms to provide an additional layer of information on which to base word alignments, before precision and recall are applied to measure similarity.

The intermediary step of computing alignments is not, however, necessary to take advantage of structural information. Amigó *et al.* [2006] calculate simple overlaps between sets of words selected using a parse tree. Their work was inspired by Liu and Gildea [2005], who adapted the concept of n-grams to apply to sequences of words or CFG nonterminal symbols which had been judged by a parser to be closely related, primarily through headword chains (paths descending through a tree) and subtrees of varying depths.

The approach of Liu and Gildea [2005] is built upon by Owczarzak *et al.* [2007a,b], who use the labelled dependencies produced by a Lexical-Functional Grammar parser [Cahill *et al.*, 2004] to produce dependency-based n-grams: primarily $(child, label, parent)$ triples. Attempting to abstract away from string-based representations of language, this parsing style allows minor rephrasings to pass unnoticed; where such detection fails, the probabilistic grammar's n most likely ('n-best') parses are all considered in an attempt to minimise noise.

Expected Dependency Pair Match [Kahn *et al.*, 2009] represents further work in this area, introducing a family of metrics which use limited or augmented information to produce more biased or varied modules. These may make use of n-best parses with varying n, or focus on different aspects of dependency relations by omitting one or more elements from the triples which encode them. Such modules can then be combined in various ways to provide a great degree of flexibility to the resulting metric.

More traditional evaluation features are built into the dependency-comparison approach by He [2010], which uses Meteor-like stemming and chunking to improve alignments and detect cohesive blocks, while also introducing extra tuning parameters and using a more well-known open-source parser [Nivre *et al.*, 2006] than its predecessors. VERTa [Comelles and Atserias, 2015], a tool combining many different evaluation approaches, includes a technique of comparison of dependency triples similar to that of both Owczarzak *et al.* [2007a] and He [2010].

Structural information from parse trees can be leveraged not just on its own, but also in the context of the original unstructured strings. SEPIA [Habash and Elkholy, 2008] compares the 'surface span' – distance between words ignoring all structural information – for pairs of words linked through dependency relations in the hypothesis and reference translations.

Unfortunately, the process of parsing reference and hypothesis sentences is inherently error-prone. The assumption that noise will be greater when processing machine-produced translations is key to the BLEUÂTRE metric [Mehay and Brew, 2006], which parses only the reference sentence using Combinatorial Categorical Grammar. It then extracts the relative orders of closely related words, and calculates recall of this ordering in the hypothesis string. Yu *et al.* [2014] build on this technique by including more varied structural elements than simple headword chains.

While the structural approaches described so far all require sentences to be parsed, the processing done on the resulting parse trees is not especially heavy-duty: this allows algorithms to be run quickly, but necessarily limits the depth of analysis. The approach of Padó *et al.* [2009] instead attempts to extract as many semantic 'features' from a dependency-parsed sentence as possible – tenses, locations, important verbs, etc. – to determine whether each of a pair of sentences semantically 'entails' the other as per Dagan *et al.* [2006].

While word-level dependency parsing is a powerful and flexible technique, more high-level representations of sentences have also been applied to evaluation. For example, HMEANT [Lo and Wu, 2011] asks humans to annotate Propbank-style predicate-argument relations [Kingsbury and Palmer, 2002] to indicate practical features of the translated sentences such as time and agency, before comparing these using F-scores. After the initial exploration to gauge the potential effectiveness of the technique, the process was automated by Lo *et al.* [2012].

Another approach to encoding the structure of a sentence is the use of discourse analysis [Joty *et al.*, 2012]. This focuses on the rhetorical elements within sentences, linked through coherence relations such as elaboration and attribution. The number of common subtrees within such structures have been compared by Guzmán *et al.* [2014] using tree kernels [Collins and Duffy, 2001], before being combined with others in the DiscoTK family of metrics [Joty *et al.*, 2014].

2.2.4 Common threads

While different metrics for evaluating machine translation can have dramatically different approaches – matching words, counting edit operations or involving semantic structure – a number of features are common between almost all. One of these is the comparison of a given machine-produced hypothesis with a human reference translation. This trend has been claimed to cause bias in human judges, impacting the quality of the evaluation [Fomicheva and Specia, 2016], although futher research has failed to confirm this [Ma *et al.*, 2017]. Some techniques attempt to bypass the need for such a reference translation [Gamon *et al.*, 2005], with the field of reference-free Quality Assessment gaining traction in more recent years [Specia *et al.*, 2010].

A second commonality between the most well-known metrics is that they all provide an overall rating: a single summary result (whether a numeric value or a grade indicator) dealing with the holistic 'goodness' of the translations. This may focus primarily on fluency or adequacy, but generally represents some attempt to amalgamate the two.

The simplicity of a single score is very useful for a number of purposes: for example, it is ideally suited to system training, or reporting the performance of a new translation tool. At least one annual event, the Workshop on Machine Translation [Macháček and Bojar, 2013] exists in part to focus on developing such techniques, and human evaluation data has been generated for both that event and others with which to judge their success [Linguistic Data Consortium, 2011]. However, useful as such holistic techniques may be, they are ill-suited to evaluation goals outside simply training a system.

2.3 Granular metrics

Given the limitations of holistic evaluation techniques, the field of more granular, error-specific evaluation techniques has appeared and is receiving ever-growing attention. Such metrics, and the greater understanding they offer about the specific strengths and weaknesses of a particular system, can be helpful in a number of scenarios.

For example, any potential application of automatic translation may focus on one feature over others. A reference manual may be relatively unaffected by inaccurate tenses but could potentially become unusable in the case of too-free use of synonyms or paraphrasings, while a weather reporter might have the opposite requirements. Similarly, knowledge of a given translation system's weaknesses could suggest specific courses of action to address these, either by working within the software or by coupling it with specific pre- or post-processing components [Popović *et al.*, 2014].

Recent work demonstrates that translation errors come in a wide variety of types [Secară, 2005], and a number of taxonomies of the errors likely to be found in automatic translation have been devised [Font Llitjós *et al.*, 2005; Flanagan, 1994]. The most commonly used is that of Vilar *et al.* [2006b].

A number of tools have been designed, after a number of years' initial exploration into the area, to provide categorised data on several types of error at once [Popović and Ney, 2007, 2011; Popović and Burchardt, 2011]. These tools include Hjerson [Popović, 2011], which uses the interactions between various error-rate tools to determine features and thus types of errors; and Addicter [Zeman *et al.*, 2011], which is based more directly on alignments, augmented through part-of-speech tags and a dedicated reordering detector. These tools have since been merged together to provide more reliable results [Berka *et al.*, 2012].

AMEANA [El Kholy and Habash, 2011] gives a smaller number of error categories than Hjerson and Addicter, but aims to be more robust when dealing with morphologically rich languages. It calculates a series of features for words – such as number or gender – then produces pairings between words in two sentences which maximise the number of matches across these features. The differences between paired words are then used to produce counts of fully matching, partially matching and unpaired words.

While investigation of errors across many categories is very useful for general meta-analysis of translation systems, it can be helpful to understand specific isolated error types in more detail than such broad tools can accomplish. When producing a tool for such error-specific analysis, the decision of which error type to investigate is far from trivial. While a handful of investigations have been performed into the relative relevance of these errors on human understanding [Lommel *et al.*, 2014; Kirchhoff *et al.*, 2012], this remains at least in part an open question.

Evidence gathered so far suggests that word ordering is the most important error type [Birch *et al.*, 2008; Isozaki *et al.*, 2010] if only for English-language output [Popović, 2012a], both from the point of view of readers and also in terms of post-editing effort [Popović *et al.*, 2014].

2.3.1 Word order

Given the apparent significance of word order in the opinions of end users on the quality of translation [Birch *et al.*, 2008], it is perhaps unsurprising that a number of metrics have been developed which investigate this feature specifically.

In some cases, word ordering is taken into account simply as one feature among several. For example, Meteor includes a second step which takes into account the ordering of those words. As discussed in more detail in Section 3.5.4, it does this by 'chunking' the sentences, finding the smallest number of groups of aligned words such that each contains words which are both adjacent and identical in both hypothesis and reference sentences. The ratio of the chunk count to the total number of aligned words is then multiplied with the tool's first calculation, an F-measure, to produce a final score. A chunking technique based on Meteor's is also presented as the 'fuzzy reordering score' of Talbot *et al.* [2011], where it is used as a standalone tool to improve translation generation.

As well as using ordering evaluation to contribute to a broader tool, it is possible to combine existing techniques to isolate ordering errors. This is the approach used by Hjerson [Popović, 2011] and earlier in a standalone tool [Popović *et al.*, 2006], which observe the differences between Word Error Rate and variants of the Position-Independent Error Rate to detect words for which only the positions have changed.

An alternative approach, that of Addicter [Zeman *et al.*, 2011], uses weighted graphs of sentence permutations with a view to detecting not just the overall severity of order errors, but the specific movements of individual words. It thus categorises ordering errors as either short-range (transposed words) or long-range.

When investigating language it is intuitively sensible to consider features of the words in question, as the above tools have done. However, by ignoring the nature of words as syntactic units and considering them only as elements of ordered sequences, a number of generic statistical techniques become applicable. For example, Kendall's τ [Kendall, 1938], designed within the field of Psychology, compares matches and differences in any two rank-ordered sequences, and has been applied to the scenario of sentence ordering [Lapata, 2003] with moderate success [Lapata, 2006]. It is described in more detail in Section 3.5.2.

More recently, it and other comparison techniques have been applied to within-sentence order comparisons. Ulam's distance [Ulam, 1972] measures the number of lateral movements required to change the order of a sequence to match another; while Hamming distance [Hamming, 1950] simply measures the number of positions in two equal-length sequences for which the symbols do not match.

Both of these, along with Kendall's τ, are investigated by Birch *et al.* [2010], with the latter two combined in the LRScore metric [Birch and Osborne, 2010]. Similarly, Isozaki *et al.* [2010] compare Kendall's τ with Spearman's ρ [Spearman, 1904], a measure of the correlation (the extent to which variation in one sequence is echoed in another) between two sequences ordered by rank alone.

While these techniques achieve sometimes impressive correlations with human judgments, they nonetheless omit an important feature of natural sentences: structure. The relevance – and, as discussed earlier, even the definition – of this factor can vary, but it can be summarised by the intuition that just as some words may be more relevant to comprehension than others, so may some phrases – or relationships between words or phrases – be more important than others. For example, order differences beyond simple transposition may be difficult to detect, while the movement of multi-word phrases may be no more detrimental to a sentence than the movement of an individual word [Stanojević and Sima'an, 2014b].

These questions have been investigated through the use of recursive decomposition of sentences into permutation trees [Gildea *et al.*, 2006], first used in the multi-technique metric BEER [Stanojević and Sima'an, 2014a] and later used to produce a standalone order evaluator by Stanojević and Sima'an [2014b].

CHAPTER THREE

PROJECT OVERVIEW

3.1 Introduction

As shown in Chapter 2, machine translation evaluation is a fast-growing, diverse field. A wide range of techniques exist for measuring the holistic quality of a sentence (Section 2.2), while a more modest group investigates specific error types (Section 2.3). Of the numerous types of mistakes machine translation systems can make, word ordering has been shown to be one of the most significant, and many metrics focus on this feature specifically (Section 2.3.1).

When measuring the holistic quality of a sentence, several metrics have investigated the relevance of sentence structure, for example through the use of dependency parsing (Section 2.2.3). Of those focusing on word order, however, to our knowledge only one form of structure – that of permutation trees [Gildea *et al.*, 2006] – has been considered thus far [Stanojević and Sima'an, 2014a,b].

We believe that the significance of word order errors is closely tied to sentence structure. Further, while permutation trees have been used to show this in terms of word alignments, our intuition suggests that more syntactic elements – such as those involved in dependency structures – may also be highly relevant. This intuition is strengthened by the positive results reported by machine translation systems which perform reordering using dependency structures [Liu *et al.*, 2010; Hadiwinoto and Ng, 2017].

Consider the sentences in Table 3.1. While the mismatch in sentence pair 2 involves a word occurring at opposite ends of the two sentences, the actual severity of that 'error' is clearly lower than that of sentence pair 3. In the latter case, the simple transposition of two adjacent words significantly obscures the message being communicated.

Most currently-existing tools discussed in Chapter 2 would be unable to respond to this disparity: indeed, most would actively assign lower error ratings to sentence pair 3. We believe that the only way to truly take into account such cases is to account for the syntactic structure of the sentence: the relationships between words.

3.1.1 Overview

To that end, we have chosen to base our evaluation of word ordering in machine translation on the use of one of the more popular structural tools: dependency parsing. Introduced in Section 2.1.1, this is discussed further in Section 3.4.3.

In our investigation of the relevance of (dependency) structure in word order evaluation, our first goal is simply to produce tools which take advantage of the former while measuring the latter. We produce two such tools.

DTED (Chapter 4) applies the concept of error rates (Section 2.2.2) to dependency trees, producing a normalised count of the operations required to turn the parse tree for a machine-produced hypothesis translation into that of a human-produced reference.

Our second tool, DERP (Chapter 5), uses a novel technique with some similarity to that of dependency label n-grams: comparison of paths. We use edit operations, calculated through Levenshtein distances, to compare the dependency relationships and surface direction of paths between aligned nodes in a reference and hypothesis translation.

To evaluate both these tools, and as a standalone contribution to the machine translation community, we have produced a database of human judgments of word order quality in translations. Named WOJ-DB (Chapter 6), this has been produced using a survey of

	Reference	Hypothesis
1	The cat sat on the mat.	The mat sat on the cat.
2	I spoke to him there.	There I spoke to him.
3	She let it be and left.	She let it and be left.
4	The keen loud dog did it.	The loud keen dog did it.
5	Then was it done.	It was done then.
6	Jill gave it to Bob.	Bob gave it to Jill.

Table 3.1: Example word order mismatches

diverse individuals in our institution's town. Each participant was asked to rate a number of both 'real' hypothesis translations and synthetically generated permutations of reference sentences. Judgments were gathered relating both to the quality of holistic adequacy and of the significance of word ordering within that.

In Chapter 7, we combine these three contributions, evaluating the content of WOJ-DB then using it to judge the success of DTED and DERP. We investigate their performance both with and without structure, in comparison with each other and with other baseline tools discussed in Section 3.5. We also respond to the overarching goals for our research, presented in Section 3.2.

3.2 Research questions

3.2.1 Pursuing accurate evaluation

When presenting a new metric to the community, one question is often considered paramount: does the tool work well? First and foremost, we would like to produce a metric which is capable of evaluating word order better than any other currently available. Thus, our first research question is:

> Can we improve on the current state of the art of word order evaluation?

Note that we are interested in predicting the assessments which a human judge would give of the *word ordering* of a sentence, rather than judging its overall quality as many other tools do. It is with this in mind that we have produced WOJ-DB, a database of judgments of this specific error type. As discussed further in Section 6.1.1, most resources currently available contain only overall quality judgments, while the few which categorise errors nonetheless fail to indicate the subjective severity of such errors. It is precisely this aspect of impact to human comprehension which we wish to evaluate.

In evaluating any metric we produce, we wish to consider two aspects of its performance. First and most important is the extent to which the scores it produces can predict corresponding human judgments. This is represented by simple correlation with such opinions. Secondly, to consider our tools truly an advance in technology such an improvement must be consistent irrespective of the evaluation environment. We consider that environment to consist of the exact parser, word alignment techniques and other tools used by, but not key to, the algorithm being investigated.

3.2.2 Relevance of structure

Having produced structure-based tools which may or may not evaluate the quality of word order to a high level, we must investigate the reasons for their performance. Given the success of the wealth of tools which rely on unstructured, surface-level information such as word n-grams or edit operations, it is important to justify the addition of the extra step of generating structural information. Thus:

> Does structure aid the evaluation of word order in machine translation?

This question remains largely unanswered in the literature, possibly as a simple result of the novelty of tools which utilise any form of structure. While the only existing example of structural evaluation of order reports encouraging results when compared with unstructured statistical techniques [Stanojević and Sima'an, 2014b], the conclusions which can be drawn from such comparisons are limited.

Simply put, the tools being compared do not work in ways which can be theoretically compared: such comparisons do not compare like with like. In a similar manner, it is difficult to extract clear implications from any comparison between our tools and existing metrics. Even if our tools perform to a higher level than a third-party unstructured evaluation, it is impossible to say that it is the use of structure which has led to such an improvement, nor that the increase would remain when compared with a different such baseline.

To truly approach the question of the effect of structure, we thus need genuinely comparable yet structure-free techniques. With this in mind, we have produced alternate versions of our two tools, DTED and DERP. In these baselines, we eliminate the structure in a manner which is transparent to the algorithm itself. The exact mechanism for this is discussed briefly in Section 3.4.4, while its implications are discussed in DTED's and DERP's respective chapters.

3.2.3 Cohesion of structure and order

While the intention behind the tools we will produce is firmly on the evaluation of word order, no aspect of the dependency structures we use necessitates this. Indeed, dependency relations between words inherently contain information unrelated to word ordering: for example, the dependency links within the phrase 'dogs and bones' are different from those of 'dogs enjoy bones' despite both matching words occurring in the same order in both phrases.

Despite this, as discussed in Section 3.1 we believe structure to be an important factor in the evaluation of the quality of word ordering. This results in two opposing predictions: first, dependency structure allows us to evaluate the word ordering of a sentence in isolation; and second, dependency structure incorporates too many non-order-related factors to permit such isolation. We thus investigate the interaction between these two predictions:

> Does dependency structure permit word order evaluation, or does it lend itself more to holistic judgment?

To respond to this question, we will need to observe the performance of our tools when judging the quality of word order, and compare that performance with that relating to overall sentence quality. We thus build both questions into our survey in Chapter 6, before reporting our results in Section 7.3.4.

3.3 Important restrictions

While our goal is to perform as comprehensive as possible an investigation into the relevance of structure to machine translation, we are nonetheless practically unable to consider every possible facet of that relationship. In practice, our experiments are restricted by three factors: the language we use, the aspect of language which we use for comparison, and the scope of our test dataset.

3.3.1 Use of English

The first and most significant restriction to our experiments is their absolute focus on English. As a result of working in an entirely English-speaking environment, without straightforward access to large numbers of native speakers of any other language, we have chosen to investigate only translation into English – although we place no restrictions on source language. As a result, any conclusions we draw about word order will be applicable only to English.

We believe this choice to be significant, both as a benefit and a limitation. Conveniently in our case, a large number of high-quality resources exist in this language, of which some are described in Section 3.4. As such, we will be able to manipulate the text we receive using third-party tools more easily, more flexibly, and with more confidence in quality than would be possible using most other languages. This allows us to work with text which has been preprocessed in any way we feel is most useful for the algorithms we devise.

The downside to the use of only one language in our experiments is that we will be unable to judge the scope of any conclusions we draw. It is well known that word order is an important factor for comprehension of English, but is less so in other languages. For example, in many morphologically rich languages the information which in English is encoded in the ordering is instead indicated through cases or other word modifiers. To speakers of such languages, the concept of ‘quality of ordering’ may be different from that of native English speakers, and may be unimportant or even meaningless.

Despite this, we believe data relating only to English is far from worthless. Beyond the obvious fact that conclusions related to English are of practical use due to the ubiquity of the language across the world, we believe that the conclusions we draw will be applicable both to other languages to greater or lesser extents, and to our understanding of structure in a language-independent sense.

3.3.2 Adequacy over fluency

As briefly mentioned in Section 2.2, evaluation of machine translation is often split into the two evaluation criteria of Fluency and Adequacy. The former refers to the extent to which a sentence uses language correctly and idiomatically, as a native speaker would, while the latter indicates how much of the meaning of the source sentence can be understood from the translation.

When producing both our evaluation metrics and the human judgments with which we will evaluate those metrics, we have chosen to prioritise just one of these two criteria: adequacy. The reasons for this are several, though the lack of consideration for both factors once again limits the conclusions we can draw from our experiments.

The most significant reason for applying this limit is a practical one: we do not consider that we have the resources to investigate both fluency and adequacy separately. Doing so would arguably require separate metrics with different design decisions, dramatically increasing the time required to produce them and also the complexity of any analysis. It would also add complexity to the evaluation we passed to human participants (Chapter 6), reducing the number of sentences which could feasibly be evaluated given the resources available.

Our second reason for choosing to measure only one of the two most popular evaluation criteria is that such an omission does not directly mean that no information is available about the ignored one. This is because the two types of assessment are inter-related, as 'annotators have difficulty drawing any meaning from highly disfluent translations, leading them to provide low adequacy scores. Similarly, for a translation to fully express the meaning of a reference, it must also be fully, or near fully fluent' [Denkowski and Lavie, 2010]. Thus, any conclusions we draw about adequacy alone may also suggest information about the quality of fluency in the translation set – although the strength of such information is unknown.

The reason why we have specifically chosen adequacy over fluency is simply related to our opinion of it as the more relevant feature. Given the arguable primary purpose of machine translation as a method of communicating a message to individuals who do not understand the source language, we consider that errors in adequacy are more harmful to this goal than those of fluency. Consider sentence pair 6 in Table 3.1, in which two words have been swapped between the two translations. While the sentence remains entirely fluent, its meaning has been dramatically changed. We intend for our metrics to detect and penalise such errors, reflecting their impact to adequacy rather than fluency.

3.3.3 Sentence-pair scores only

The final restriction we place on our metrics is to design them to produce scores only for individual sentence pairs. This is a much weaker limitation than those described above, as it does not in itself prevent us from producing meaningful evaluations which fully address the questions we have put forward in Section 3.2. It does, however, prevent us from following two common trends in machine translation evaluation: first, we do not take into account multiple reference translations for a single hypothesis; and second, we do not produce system-level scores.

A large body of research exists which suggests that evaluation using multiple reference translations can produce better results [Papineni *et al.*, 2002; Fomicheva and Specia, 2016]. The intuition behind this is that most sentences can be translated in a variety of ways, so comparison metrics like ours run the risk of penalising certain hypotheses simply because, for example, they prioritise different aspects of the source, even if both approaches are valid.

For example, consider the Polish phrase “Zakasał rękawy”, which literally translates to “He rolled up his sleeves”. While the literal translation is a valid idiom in English, depending on context an entirely legitimate translation could equally be “He got ready to work” – which would however be considered by many automatic metrics to be a dramatically different (and thus incorrect) sentence. By providing multiple reference translations, we reduce the likelihood of such a mismatch occurring.

While sentences with multiple reference translations can provide reliability by increasing the information available when scoring an individual sentence, another approach to ensuring reasonable judgments is simply to produce scores relating to an entire system at once.

Such scores are based on the assumption that virtually any automated algorithm will produce an unintuitive or unreasonable score in some situations, as a simple consequence of the enormous complexity and diversity of natural language. Further, it is natural to assume that some sentences’ qualities will be overestimated, while others will be underestimated.

If the scores for all sentences translated by a given system are aggregated using a simple technique such as the arithmetic mean, it is hoped that such inaccuracies will be ‘smoothed out’. As a result, the ‘system-level’ score is expected to be a more reliable measure of that system’s translation quality than a measurement based on any individual sentence.

This expectation is strongly vindicated in practice: when automatic judgments are compared with humans’ using various techniques, system-level correlations of 0.7 or more are commonly reported while segment-level correlations are often close to 0.3 [Lavie and Denkowski, 2009; Fishel *et al.*, 2012b; Stanojević and Sima’an, 2014a]. While this may in part simply be due to the sparsity of data involved in many system-level correlations, it is nevertheless a powerful trend.

An additional benefit of the use of system-level scores is their simplicity when training a system. A common procedure in machine translation is to run a metric such as BLEU, alter the translation system in some way then run the same metric again on the output of the updated system to determine whether the change resulted in an improvement [Och, 2003]. While scores describing the entire system are easy to use in such a situation, scores for individual sentences are not directly relevant.

Given these benefits of system-level and multiple-reference scores, why then have we limited ourselves to simple sentence pairs? The reasons are to do with the practicalities of our evaluation environment, or more specifically a consequence of the fact that we are attempting to produce judgments which are not a standard of the machine translation community. As discussed in more detail in Chapters 6 and 7, we have chosen to produce our own judgments on word ordering specifically.

Without the resources of more major evaluation environments such as the Workshops on Machine Translation [Bojar *et al.*, 2014, 2015, 2016a], our database of judgments is limited in scope, with a total of 1783 sentences scored. While adequate for our purposes when considering sentence-level scores, the separation of such scores into individual systems – based either on the real translation tools used or on more synthetic divisions produced by, for example, bootstrap resampling [Stine, 1989] – would result in ‘systems’ containing too few datapoints from which to draw reliable conclusions.

Similarly, we have elected not to include multiple references in our experiments for two reasons. The first is due to the source of our sentences, discussed in Section 6.1.1: the shared tasks of the Workshops on Machine Translation, while providing diverse and plentiful translations, do not incorporate the use of multiple reference translations. The second reason is an assumption that incorporating multiple reference translations in the survey we used to collect our human judgments (Chapter 6) would have overly complicated it from the point of view of our non-expert participants.

While the limitation to sentence-pair scores prevents our tools from being as broadly applicable as they might otherwise be, we do not consider them to pose a severe problem. This is primarily because such a feature would, in our view, be considered part of the process of fine-tuning and perfecting an approach. In practice, given the significant lack of existing structural tools in word order evaluation, our own are intended primarily as proofs-of-concept, demonstrating the validity of their approach rather than aiming to be the last word in the area.

Additionally, note that the above reasons for not incorporating system-level or multi-reference scores are related to our experimental setup, not our algorithms themselves. Indeed, should system-level scores be desired in the future, their addition is a simple process. Various techniques have been proposed in literature, such as the logarithm-based geometric mean of BLEU (see Section 3.5.3), but one of the most common and simplest is a simple arithmetic mean of all sentence-level scores.

Similarly, support for multiple reference translations can be added without significantly altering the core algorithms we present. This could be done in the same manner as Meteor and others [Giménez and Màrquez, 2007], where "If more than one reference translation is available, the translation is scored against each reference independently, and the best scoring pair is used" [Lavie and Agarwal, 2007].

3.4 Tools used

In our investigation of structure in word ordering, we have produced two tools: DTED and DERP. Both of these tools leverage the dependency structure of their input sentences as a core part of their algorithm. However, as discussed in Section 2.1.3, producing such structures is itself a far from trivial problem with decades of research behind it.

In order to parse any real translated sentences, they must undergo a number of pre-processing steps. Two of these are common among all types of natural-language parsers: *tokenisation* and part-of-speech *tagging*. A third is more specific to the area of machine translation: given two related sentences – in our case, a machine-produced and human-produced translation of the same source sentence – *word alignment*s are generated to indicate which words are equivalent to each other.

For tagging, word alignment and parsing we have chosen to apply multiple tools of each kind. This is in order to better investigate the effects of our algorithm in isolation: given that both DTED and DERP directly utilise the output of the parser and aligner, with the parser relying on our part-of-speech tagger, it is straightforward to imagine that the accuracy of all three utilities will have knock-on effects on the overall result.

Somewhat more concretely: should a given phrase be misunderstood by the parser or aligner, leading to a parse which does not well represent the structure of the sentence or an overzealous or particularly sparse alignment, any comparison on that sentence is likely to suggest highly unpredictable – and relatively meaningless – conclusions.

This knock-on inaccuracy is impossible to entirely eliminate without provably perfect tools, but can be mitigated by providing alternatives for them. This could for example be by combining a series of scores produced using various tool combinations to produce an average score: such a score would to a certain extent abstract away from any one tool.

However, while such an aggregate score would be useful, we considered it more interesting to investigate the exact extent to which our scores vary based on the utilities used. Such variability is a reflection on the reliability of the tool, and can at least be approximated by comparing the efficacy of scores from DTED and DERP with different third-party tool configurations. Further discussion of such evaluation can be found in Chapter 7.

One more reason to build DTED and DERP to support multiple 'configurations' is that this avoids the need for us to rely on an individual project or package for our metrics to work. Should a tool we use be discontinued for any reason, our own work retains its usefulness as a replacement can be provided. Potentially more importantly, should a new piece of software be created which produces more accurate information for our own than any of those available when ours were designed, that tool can be integrated easily and may result in improved performance.

We have chosen to use exactly two taggers, parsers and aligners in our experiments. The choice to use only two is partly for practical reasons, as we did not have the resources to incorporate even a representative subset of the wide array of tools available today. Additionally, we believe that alternatives for each of three utilities – thus resulting in at least six configurations for our tools – provides enough scope to observe with some measure of confidence the variability in our tools' performance.

3.4.1 Tokenisation

Before the parser, tagger and aligner can be run, the first step in processing our text is to determine which atomic units are to be parsed. This is the process of tokenisation: splitting sentences into individual words or other separate units. Often, separations between these can be complex or ambiguous to detect: for example, which of the following should be split into multiple words? "light-headed" – "often-considered" – "13 452" – "13,452" – "you know" – "y'know" – "y,know".

While detailed context-dependent tokenisation has been shown to improve translation [Zalmout and Habash, 2017], in our project we have chosen to use only one simplistic preprocessor. This is because we do not anticipate that the variations caused by different ones will impact the fundamental results of our experiments to the same extent as the three processes discussed above and described below.

Given this, we have in all cases applied to our input the tokeniser packaged with the well-known Europarl project [Koehn, 2005], which aims to make available large quantities of translations taken from the European Parliament proceedings. Specifically, we have used the tokeniser included in the Europarl v6 Preprocessing Tools [Koehn and Schroeder, 2011]. We chose this system due to its free availability and widespread use.

3.4.2 Tagging

While tokenisation is a simple problem made complex by occasional edge cases, part-of-speech tagging is a much more challenging problem at all levels. Again, many words are ambiguous, with different parts of speech being applicable depending on contextual features which can be hard to access computationally.

Various approaches have been taken in the past, with many earlier systems relying – as with most areas of NLP – on sets of rules crafted either by human experts or automatic processes. One of the most well-known such systems is the Brill tagger [Brill, 1992], with error rates of under 5% reported.

However, more recently purely stochastic systems have achieved success, such as those making use of the information-theory model of maximising entropy [Ratnaparkhi, 1996]. These stochastic tools rely on existing corpora of pre-tagged text, from which rules are learned in a manner closely related to grammar inference (Section 2.1.2).

For our project, we have chosen two widely used and easily accessible tools, one of which is a key part of a major ongoing academic project while the other is a simpler library implementation of a traditional technique. The latter system was written for Python's wide-ranging NLTK toolkit [Bird, 2006], which offers several open-source tools for part-of-speech tagging. For simplicity, we have chosen an implementation of the technique proposed by Ratnaparkhi [1996], referred to simply as the MaxEnt parser throughout this document [Malecha and Smith, 2010].

The second tagger we use is part of the Stanford CoreNLP project [Manning *et al.*, 2014], a significant academic endeavour which includes a variety of language-processing tools. These include a standalone part-of-speech tagger which is also based on the maximum entropy principle [Toutanova *et al.*, 2003], but which incorporates bespoke additions to improve the treatment of unknown words, verb form disambiguation and particle disambiguation. We have used the implementation and models bundled with the project's dependency parser v3.7.0 [Klein and Manning, 2003b].

Note that a fundamental element of the process of part-of-speech tagging is the choice of tags used. In stochastic taggers such as our maximum-entropy-based utilities, this is extracted as part of the training process. Both of the taggers we have chosen have been trained on, and thus use the tagset provided for, the well-respected and long-running Penn Treebank project [Santorini, 1990].

3.4.3 Dependency structure

Once input sentences have been tokenised and those tokens assigned part-of-speech tags, they can be parsed to produce the dependency structures we rely on. These consist of trees in which each edge indicates a semantic link between two words. For example, a noun may be the subject or direct object (NSUBJ or DOBJ) of a verb, and may itself be qualified by a determiner (DET) or an adjectival modifier (AMOD). The root of a dependency tree is an important verb in its sentence.

While a number of possible syntactic frameworks exist, as discussed in Section 2.1.1, we have selected dependency structures for a number of simple reasons. First, like context-free structures, dependency trees are able to represent complex linguistic phenomena using relatively lightweight grammars and quick parsers. The considerations of size and speed are far from trivial, affecting how practicable any evaluation tool is in the notoriously data-intensive field of translation.

A second relevant feature of dependency parsing is its goal of capturing the key semantic relations between words in parsed sentences. While some other formalisms such as CFGs attempt primarily to describe the *nature* of words and phrases in a sentence, dependency structures attempt to encode the *purpose* thereof. We believe that this perspective is more closely tied with the problem of translation evaluation: two translations should contain elements which, while not necessarily exactly matching in nature, perform the same functions.

Our decision is in keeping with the general trend of translation evaluation, with several existing tools relying on the technique as discussed in Section 2.2.3. However, should another formalism be considered relevant to the techniques we employ, they could be adapted without excessive work: nothing in our tools is inherently tied to dependency structures.

Visual representation

A number of representations of dependency trees exist in literature, including purely textual representations [He, 2010], linear textual sentences with arcs to indicate dependencies [Clark *et al.*, 2002; Nivre, 2003], visual tree structures akin to those used for context-free grammars [Hajič *et al.*, 2012] or even combinations of these [Gómez-Rodríguez *et al.*, 2011].

In the various examples throughout this document of tree structures, we have chosen to use the structure exemplified in Figure 3.1. A dependency label, indicating the nature of a dependency relation, may be shown as a string of grey capital letters adjacent to a black line between the parent and child of that link. Such parents and children may be shown in black to indicate that they are not aligned with any word in another tree, while such alignments – introduced in Section 3.4.5 and first shown in Chapter 4 (page 45) – are indicated through matching colours.

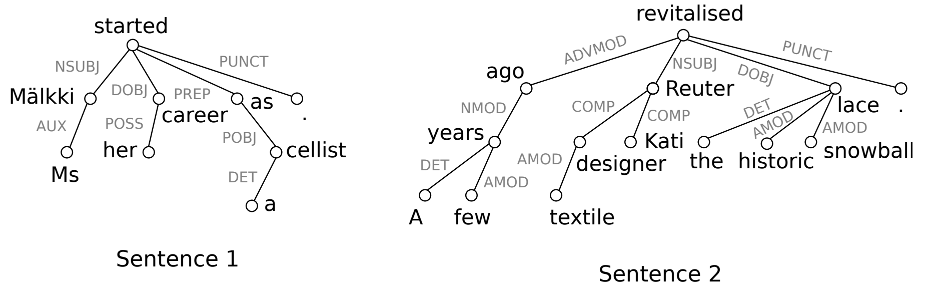
A number of the examples we use are based on sentences used within our experiments (Chapters 6-7). This is the case, for example, with the two examples shown in Figure 3.1. However, occasionally we have constructed synthetic example sentences or phrases, intended to highlight certain aspects of the tree(s) or algorithm(s) in question.

The original text from which a dependency parse tree was produced can be read directly from such figures. This is done by considering the nodes (words) strictly from left to right, irrespective of vertical position within the tree. For clarity, original sentences are also shown underneath the visual tree structures in most cases.

Note that while dependency labels are included in all the parse trees we use, they are only core to the functionality of one of our tools. DTED (Chapter 4) discards the information they contain, and consequently all figures in its chapter omit the dependency labels from any parse trees shown.

Projects used

To generate the parse trees we use in our experiments, we have used two well-known tools, chosen both for their high quality and popularity in other projects and for their opposing and thus complementary approaches to parsing. We were able to choose from



Sentence 1

Sentence 2

Figure 3.1: Two sample dependency trees. Dependency labels are distinguished from words by grey capital letters.
Sentence 1: Ms Mälkki started her career as a cellist.
Sentence 2: A few years ago textile designer Kati Reuter revitalised the historic snowball lace.

a number of high-profile dependency parsers which are available, including the Malt parser [Nivre *et al.*, 2006], the Stanford Parser [Klein and Manning, 2003b] and the Berkeley parser [Petrov and Klein, 2007].

Each of these represents ongoing long-term collaborative projects, with numerous releases over several years. Each is separated into two primary components, with the main executable typically governing both the parsing of a given sentence and the generation of the second component: information specific to the language being parsed, usually produced using the techniques mentioned in Section 2.1.2. Such models can either be obtained ready for immediate use from the same sources as the parser itself, or can be generated by the end user from a bespoke treebank.

We have chosen to use the first two tools mentioned above: the dependency parser implemented in the MaltParser framework [Nivre *et al.*, 2006], and that produced as part of Stanford CoreNLP [Manning *et al.*, 2014].

The Stanford Parser was originally built as an unlexicalized probabilistic context-free grammar parser [Klein and Manning, 2003b], intended to demonstrate the viability of the unlexicalized approach. This involves the parser annotating phrasal subtrees specifically according to function words (*for*, *to*, etc.) rather than simple head nodes: those which have been considered to best represent the nature of a subtree.

The Stanford Parser was then extended to allow conversion from the original phrase-structure trees to dependency relations [de Marneffe *et al.*, 2006]. This is done in two steps: first the semantic head of any given subtree is calculated, often different from the syntactic head produced in the original parse. Next, the types of relations between heads and their erstwhile phrasal siblings are calculated using pattern-matching techniques.

The approach of the MaltParser is rather different from that of the Stanford project, focusing on deterministic shift-reduce parsing rather than the more common probabilistic approach. Unlike the Stanford system the MaltParser does not rely on a grammar *per se*, instead using a series of learned mappings from parser states to appropriate actions for the parser to take. Nonetheless, both forms of linguistic data are learned from gold-standard treebanks, with both of the systems we use having been trained on the highly popular Penn Treebank [Marcus *et al.*, 1993].

We believe that these two systems represent reliable and high-quality approaches, ensuring that the parses we use are as legitimate as possible. They are, however, different

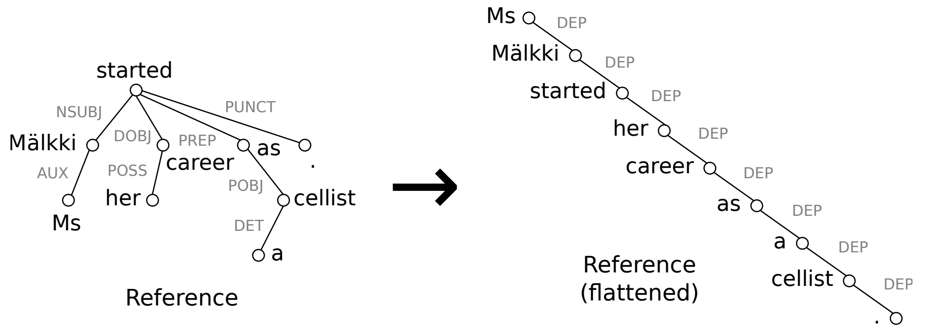


Figure 3.2: Original and flattened versions of a sample dependency tree

enough from each other to allow for variety in those parses inasmuch as is permitted by the individual sentences. We are thus confident that they will, as intended, allow us to observe the effects of different parses on our tools.

3.4.4 Flattened dependency 'structure'

As mentioned in Section 3.2.2, in order to provide a reference case to deepen our understanding of the importance of structure to our tools, we run them both on parsed dependency trees and on flattened versions of those trees. To produce these flattened structures, we simply run a preprocessing step before executing the evaluation algorithm in question as normal.

In actual fact, the flattening process does not eliminate all structure from the sentences: it merely forces the trees to contain purely linear information information by applying a uniform rule to that structure. Specifically, it forces each node to be the (only) immediate child of its predecessor, as shown in Figure 3.2. This is done to allow a representation as close as possible to the unprocessed, unstructured sentences while still being in the format expected by our tools.

This process is inherently wasteful, as it requires real parse trees to be produced then immediately discarded. This inefficiency could be resolved by producing flat parse 'trees' directly from string-form sentences, resulting in much quicker runtimes for our tools.

The primary reason why we have not done this is for simplicity. Note that our project represents only an initial attempt to understand the relevance of structure to machine translation evaluation. As such, we do not consider it critical that all elements be optimised in aspects like speed of execution, which do not contribute to the feature we are investigating – namely their ability to predict human judgments. We also do not expect the flattened versions of our tools to perform better than the structured versions: as such, we expect the former to be primarily of use within this project, with little need for standalone, optimised implementations for further use.

3.4.5 Word alignment

While parse trees are the primary input required by our tools, they are not the only one. Both DTED and DERP require supplemental information on which of the words in the input sentences correspond: the word alignments between the input sentences.

There are two primary avenues we could explore for the extraction of such alignments. That taken by a number of existing metrics is to pair words according to simple techniques such as exact or word-stem matching, as shown in Sections 3.5.3 and 3.5.4. These can be used with little postprocessing [Papineni *et al.*, 2002; Banerjee and Lavie, 2005], or with statistical disambiguation techniques to provide more fine-grained mappings in cases like repeated words [Zeman *et al.*, 2011].

The alternative to such metric-specific methods is to apply an existing project designed to deal with word alignment in a broad context. A number of these exist, including the standalone GIZA++ [Och and Ney, 2003] and the Berkeley Aligner [Liang *et al.*, 2006], and that of the cdec project from Carnegie Mellon University [Dyer *et al.*, 2010].

Such ready-made alignment tools are generally intended for the treatment of sentences in different languages, as part of the machine translation pipeline. They thus use hugely more complex techniques than the aligners built into metrics, as they cannot rely on the words they align having any surface features in common at all.

Interlingual alignment tools generally rely on statistical methods to train language models, not entirely dissimilarly to both parsing and part-of-speech tagging. These can rely on Hidden Markov Models (HMMs) [Rabiner, 1989], assuming the word alignment to be a hidden set of mappings used by the translation process [Vogel *et al.*, 1996], which must be estimated through probabilistic techniques such as the EM algorithm [Dempster *et al.*, 1977].

Other techniques for interlingual word alignment can involve heuristics, based on matrices built from observations of co-occurrence of pairs of words [Melamed, 2000]. These methods are considerably simpler than the more generic statistical training, but have been shown to perform somewhat less well in practice [Och and Ney, 2003].

Purpose-built interlingual alignment systems have the benefit over simpler monolingual techniques that they consider all types of word matchings which occur in their training database, rather than relying on linguistic features which may or may not exist. Consider the single-word sentences “Alright” and “Sure”: these are both legitimate translations of the French phrase “D’accord”, yet no word-stemming technique can be reasonably expected to equate them. This could only be done through the use of external information, whether statistically learned or provided through external databases such as WordNet [Miller, 1995; Banerjee and Lavie, 2005].

Given the complexities involved in the field of word alignment, both metric-internal and standalone alignment tools impose restrictions on their mappings for the sake of computational tractability. The majority of metrics we inspected contained a strict limitation: no word in either sentence can be paired to more than one in the other [Lavie and Agarwal, 2007; Zeman *et al.*, 2011]. While many off-the-shelf systems impose the same restriction [Melamed, 2000], GIZA++ and cdec’s alignment module both relax this assumption somewhat, permitting words in one sentence only to be unidirectionally aligned to multiple in the other. In our experiments, we allow hypothesis words to be aligned to multiple reference words, but not the reverse.

Given all the above considerations, we have chosen to use two open-source off-the-shelf systems intended for multilingual processing: cdec’s alignment module and GIZA++. Both use Hidden Markov Models, with cdec generating an estimated best match between parses of both input sentences [Dyer, 2010]. GIZA++ augments traditional HMM techniques with a number of word-fertility models [Brown *et al.*, 1993], in its effort to incorporate a variety of proven techniques.

These two tools allow us the flexibility of limited one-to-many alignments, such as "was" being aligned to "has been", and the genericity of their language-independent approaches. As mentioned earlier, their alignments could be replaced by those of other techniques with very little effort if desired.

3.5 Baseline tools

As discussed in Section 3.2, we have a number of goals for our experiments. One of these is to evaluate the relevance of structure to our tools, for which investigation we have generated versions of our tools based on structured and flattened sentences. However, a simpler and highly relevant concern is whether the tools perform to a reasonable standard under any configuration.

To determine this, we will need to have an understanding of what a 'reasonable' standard is. Happily, a range of tools already exist to measure both general quality of translation (Section 2.2) and specifically word order (Section 2.3.1). Additionally, very simple inspection of the sentences we use can give us a more direct view of their characteristics.

The simplest baseline 'metric' we apply is that of simply calculating the percentage of words which are aligned in each sentence pair. Alongside this we use Kendall's τ [Kendall, 1938], which has been used in a number of investigations into the quality of word ordering in particular [Lapata, 2006; Birch *et al.*, 2010].

Of the myriad bespoke translation-evaluation systems introduced in Chapter 2, we have chosen two of the most popular as the baselines we will use in our investigation: BLEU [Papineni *et al.*, 2002] is one of the most well-known and used metrics for holistic evaluation available; while Meteor [Banerjee and Lavie, 2005] is a popular tool which evaluates holistic quality but contains a distinct ordering component.

Throughout this section, we will provide examples showing exactly how these tools function in the evaluation environment. For simplicity and comparability, we will use the same sentence pair for all of these examples: a variant of sentence pair 1 from Table 3.1, as follows.

> **Hypothesis:** the mat had sat on the cat
> **Reference:** cats had sat on a mat

Note that as we attempt to be consistent with the original papers introducing the respective tools, our terminology will slightly change when discussing each one.

3.5.1 Aligned Percentage

Our first and simplest baseline tool is an inspection of the number of pairs of aligned words among the reference and hypothesis sentences. The fundamental assumption behind this 'metric' is that the more similar a hypothesis sentence is to a reference, and thus the higher quality it is, the more words are likely to be shared between the two.

This is obviously a flawed assumption: legitimate elements such as synonyms and paraphrasings may result in falsely low scores, while matching words with incorrect features such as ordering can cause the aligned percentage to be an overestimate of quality. Nevertheless, we consider it to be an interesting baseline.

Word	Concordant with	Discordant with
mat	*none*	cats, had, sat, on
had	sat, on	cats, mat
sat	had, on	cats, mat
on	had, sat	cats, mat
cat	*none*	had, sat, on, mat

Table 3.2: Edit operations calculated by Kendall's τ on the example sentence pair

Most relevantly, both of our bespoke metrics rely to a great extent on the alignment between words: as detailed in the chapters for DTED and DERP (Chapters 4 and 5 respectively), word alignment is one of the key inputs to both metrics. Given this, we consider it important to understand the extent to which the alignment alone can account for any success we observe: this allows us to observe the improvement due to our algorithm alone, similarly to how unstructured versions of our tools allow us to observe the effect of structure alone.

We have run both GIZA++ and cdec (Section 3.4.5) on each sentence and returned the total proportion of aligned words (H_{ali} and R_{ali} in the hypothesis and reference translation respectively) relative to all words in those sentences (H_{all} and R_{all} respectively) according to the following formula:

$$alipc = \frac{H_{ali} + R_{ali}}{H_{all} + R_{all}} \tag{3.1}$$

For our example sentences, the following sets of alignments are possible. Note that due to the necessity of training the two alignment systems we use on large corpora, and the fact that our example sentences do not occur in those we use, we have not used GIZA++ or cdec to produce these alignments: they are examples only. Words from the hypothesis sentence are shown first.

Alignment 1: (cat, cats), (had, had), (sat, sat), (on, on), (mat, mat)
Alignment 2: (cat, mat), (had, had), (sat, sat), (on, on), (the, a), (mat, cats)

These alignments result in scores of $10/13 \approx 0.769$ and $12/13 \approx 0.923$ respectively.

3.5.2 Kendall's τ

A somewhat more complex statistical technique than simply observing aligned words is that of Kendall's τ [Kendall, 1938]. Designed as a correlation coefficient for evaluating ranked data, it counts pairwise comparisons between such ranks. Given two ordered sequences, such as series of numbers, the number of *concordant* pairs is considered to be the number of pairs of entries which occur in the same relative order in both sets, while *discordant* pairs are the opposite. These are compared against the total number of possible matches using a simple mathematical formula:

$$\tau = \frac{|pairs_{concordant}| - |pairs_{discordant}|}{|pairs_{concordant}| + |pairs_{discordant}|} \tag{3.2}$$

The range of this formula is the same as that for many, if not all, correlation coefficients: a correlation of 1 indicates that the order of all pairs of entries match – i.e. the ranks are identical – while a correlation of -1 indicates that the ranks in one sequence are the exact inverse of those in the other. Intermediate values denote more limited similarity, with a correlation of 0 indicating that as many pairs match as do not match, suggesting that the two sequences are entirely independent of each other.

Note that the coefficient does not in any way account for the absolute identities of, or differences between, elements of the sequences: it merely compares their relative ordering. Thus, $(3, 5)$ is concordant with $(1, 2)$, $(3, 5)$ and $(1, 7)$ due to shared monotonicity, while being discordant with any decreasing sequence such as $(7, 1)$.

Algorithm 1 Kendall's τ wrapper

```
procedure TAU WRAPPER($a_{pairs}$)
    $a_{hyp} \leftarrow$ [ hypothesis indices from $a_{pairs}$ in ascending order ]
    $a_{ref} \leftarrow$ [ reference indices from $a_{pairs}$ in ascending order ]
    $a_{mix} \leftarrow$ [ ]
    for all $i_{ref}$ in $a_{ref}$ do                          ▷ Process each index in $a_{ref}$ in order
        $i_{hyp} \leftarrow$ [ hypothesis indices which map to $i_{ref}$ in $a_{pairs}$]
                                                     ▷ Replace reference index with hypothesis one
        $a_{mix} \leftarrow a_{mix} + i_{hyp}$               ▷ Append hypothesis index to new ‘reference’ list
    return $\tau(a_{hyp}, a_{mix})$                    ▷ Calculate Kendall's τ on the resulting indices
```

While intended for generic rank comparisons, Kendall's τ applies almost directly to the field of translation order evaluation. It has in the past been used to evaluate other systems' success in ordering tasks [Lapata, 2003], and more recently has been applied directly to the evaluation of sentences [Birch *et al.*, 2010].

The key adaptation required when using Kendall's τ in our field, as introduced in Section 2.3.1, is to simplify full sentences into comparable ordered sequences. To do this, we use a very simple bespoke technique. Once again we make use of word alignments generated by third-party tools. These produce series of paired indices for equivalent words, which can then be converted to comparable sequences before calculating a normal τ value representing the order similarity of the two sentences.

The simple conversion process is shown in Algorithm 1: from the pairwise mappings provided by alignment tools (e.g. *1-1 1-2* to indicate hypothesis word 1 being aligned to reference words 1 and 2), both sets of indices are extracted separately. Indices referring to reference words are replaced with those of the corresponding hypothesis word(s) to ensure like is compared with like, though the reference words' order is retained. Kendall's τ is thus applied to the hypothesis indices in the order they occur in the hypothesis tree, and the same indices in the order their counterparts occur in the reference tree.

Note that the [-1,1] range of Kendall's τ is not entirely in keeping with the norm for machine translation evaluations, which range instead from 0 to 1. We do not consider this to be a problem, as the relative ranks of two sentences can still be meaningfully interpreted, with a lower score in all cases representing a lower level of similarity than a score closer to 1. Fundamentally, while in many statistical domains a negative correlation can indicate a useful effect – e.g. time to reach a destination vs. speed of transport – we consider that in the case of relative word ordering a negative correlation is simply worse than no correlation at all.

Component	Calculation	Result
r	length of reference	6
c	length of candidate	7
N	maximum n chosen	3
All weights w_n	$1/N$	0.33
Precision with $n = 1$ (p_1)	$4/6$	0.67
Precision with $n = 2$ (p_2)	$2/5$	0.4
Precision with $n = 3$ (p_3)	$1/4$	0.25
Brevity penalty (BP)	$\begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases}$	1
BLEU	$\text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \text{log} p_n\right)$	0.322

n	Reference n-grams	n-gram matches
1	cats / had / sat / on / a / mat	had / sat / on / mat
2	cats had / had sat / sat on / on a / a mat	had sat / sat on
3	cats had sat / had sat on / sat on a / on a mat	had sat on
4	the cat sat on / cat sat on the / sat on the mat	*none*

Table 3.3: Calculations performed by BLEU on the example sentence pair

Table 3.2 shows which pairings are considered correct or incorrect in our example sentence pair. Note that each pair is listed twice, in the rows corresponding to each of its two components. There are thus 3 matching or concordant pairs of words, and 7 mismatching or discordant ones. We can pass this information to Equation 3.2 to find a final score for the sentence pair of $\frac{7-3}{7+3} = \frac{4}{10} = 0.4$.

3.5.3 BLEU

Introduced in Section 2.2.1, BLEU [Papineni *et al.*, 2002] has been the *de facto* standard for the evaluation of machine translation for many years. While many metrics have been produced which either improve on BLEU's techniques or provide their own (Sections 2.2-2.3), it has remained popular, arguably due to its speed of execution and language agnosticism.

In an attempt to represent real trends in language corpora, BLEU is split into components, each calculable through simple equations. First, the modified precision is calculated separately for each n-gram length. Second, an exponential brevity penalty is calculated based on the relative lengths of the reference and candidate (hypothesis) sentence.

Next, the various n-gram precisions are combined, using logarithms to capture the non-linear decay the authors found in the number of n-grams of higher orders: while 1-grams (individual words) in a candidate translation are likely to be found somewhere in the reference translation(s), it becomes less and less common that any longer phrases be found in exactly the same form in both.

Finally, the brevity penalty is combined with the n-gram precisions to produce a final BLEU score for the given candidate. Note that while this score is intended to increase or decrease with the quality of the sentence, it does not itself indicate any absolute quality. Indeed, the original authors caution against any non-comparative reporting of scores, as these can be greatly influenced by non-qualitative factors such as the number of reference translations used [Papineni *et al.*, 2002].

In our example, we make two significant simplifications from the general use-case for which BLEU was designed. First, BLEU was intended to permit the use of multiple reference translations at once, for more reliable scoring. The variations between the standard definition of precision and the modified form that they use are related to this possibility, and for possible duplication of words within individual sentences. We have ignored these alterations, and use the standard definition of precision. This does not result in any difference from scores produced by genuine BLEU, as we provide only a single example for which all words in each sentence are unique: it merely simplifies the equations we present.

Secondly, a common use of BLEU is to be applied to an entire corpus at once, rather than individual sentences. This is used in the original paper as a justification for certain inaccuracies due to variations in phrasing, and most importantly minimises an unfortunate side-effect of the BLEU formula. If no n-grams match for any n included in its calculation, the choice of combining precisions through logarithms (equivalent to the use of a geometric mean) results in flat resulting scores of 0 [Banerjee and Lavie, 2005]. As such, while n may be up to any value in practice, we have limited it to 3 in our example due to the lack of matching 4-grams.

The n-grams found within our example reference sentence, and all calculations based on them including the final BLEU score for the sentence pair, are shown in Table 3.3.

3.5.4 Meteor & Meteor (Chunking)

For our second purpose-built machine translation baseline metric, we have chosen Meteor [Banerjee and Lavie, 2005]. Also introduced in Section 2.2.1, Meteor has a convenient property which makes it especially relevant to our comparative study. It is primarily intended as a holistic tool, evaluating word order quality in general by attempting to predict a combination of human judgments of fluency and adequacy – yet contains an isolatable component which is intended to produce a score multiplier based on word order quality specifically.

The standalone nature of the second phase of Meteor's pipeline means that we can extract it and consider it an existing metric for word order, albeit not one which has been formally published as such. We have thus modified Meteor trivially to ignore the initial F-measure and return a score based only on the penalty produced by its chunking component; both the off-the-shelf holistic tool and the standalone chunking component have been used for the experiments we discuss in Chapter 7.

The exact functionality of Meteor is shown through our example sentence pair in Table 3.4. Note that many variations on Meteor have been produced [Lavie and Agarwal, 2007; Lavie and Denkowski, 2009; Denkowski and Lavie, 2011, 2014]. In our experiments we use version 1.5 [Denkowski and Lavie, 2014], which contains extensions re-

Component	Contents	
‘exact’ matches	(mat, mat), (had, had), (sat, sat), (on, on)	
‘stem’ matches	(cat, cats)	
‘WN synonymy’ matches	*none*	
$unigrams\_matched$	mat / had / sat / on / cat	
$chunks$	mat / had sat on / cat	
	Calculation	Result
s	unigrams in system translation	7
r	unigrams in reference translation	6
m	matching unigrams	5
P	m/s	0.714
R	m/r	0.833
$Fmean$	$\frac{10PR}{R+9P}$	0.843
$Penalty$	$0.5 * \left(\frac{\#chunks}{\#unigrams\_matched}\right)$	0.3
$Score$	$Fmean * (1 - Penalty)$	0.590

Table 3.4: Calculations performed by Meteor on the example sentence pair

lated to paraphrase matching, parameter tuning and more. However, for simplicity in our example we strictly represent the original version of the metric presented in 2005.

The first step is to produce unigram mappings (word alignments) according to three separate techniques. These are indicated using the same visual convention as in Section 3.5.1. First, all *exact* matches between words are detected. Next, any remaining words are matched if they have a common *stem*: if they are derived from the same base word. Lastly, if any as-yet-unpaired words are indicated to be synonyms according to the WordNet resource [Miller, 1995], they are matched by the *WN synonymy* module.

From these mappings, precision (P) and recall (R) can be calculated according to their standard formulae. These are then combined using a harmonic $Fmean$ weighted strongly towards recall, as such a weighting was found to outperform more egalitarian measures in an earlier study [Lavie *et al.*, 2004]. Later versions of the tool treat the relative weight of precision and recall as a parameter which is empirically tuned [Lavie and Agarwal, 2007].

Meteor accounts for multi-word phrases through its $Penalty$, which is calculated according to the minimum number of $chunks$ into which the matched unigrams can be split. Each chunk is composed of one or more mapped words which are adjacent in both the system (hypothesis) translation and the reference. In our example, only one chunk longer than a single word can be extracted from the middle of the sentence; however, this chunk spans more than half of the sentence, producing a medium-severity penalty.

Finally, the $Fmean$ and $Penalty$ are combined to produce an overall Meteor score for the sentence pair.

Chapter Four

DTED: Dependency-based Tree Edit Distance

4.1 Introduction

Our first foray into the creation of tools to evaluate machine translation word ordering using dependency structure is an adaptation of a solution to a problem broader than our domain. The comparison of tree structures is a well-known problem within Computer Science, highly relevant to processing of documents consisting of tree-like structures such as XML or HTML [Pawlik and Augsten, 2011]. When synchronising such documents, performing searches for similar documents, or performing a host of other such comparison tasks it is essential to have a numerical representation of the difference between trees.

Such representations can take various forms, as explored in Section 2.2.3. Two of the most high-level include Tree Kernels and Tree Edit Distance [Guzmán *et al.*, 2014], the former of which has already been used in the DiscoTK metrics [Joty *et al.*, 2014]. We have chosen to apply the latter, due to its theoretical similarities with the common approach of error rates as discussed in Section 2.2.2.

First introduced by [Tai, 1979], a tree edit distance is a count of the actions required to convert one ordered tree into another. Similarly to the unstructured techniques discussed earlier, these are: Renaming or Deleting an existing node, or Inserting a new one. These actions are demonstrated in Figure 4.1, with an example of a real sentence in Figure 4.2.

A number of variants on this model have been proposed, many attempting to improve the efficiency of the algorithm when applied in large-scale or high-throughput areas [Bille, 2005]. The algorithm we have implemented is an extension of that proposed by Demaine *et al.* [2009], which is worst-case optimal, running in $O(n^3)$ time where n is the number of words in the shorter sentence.

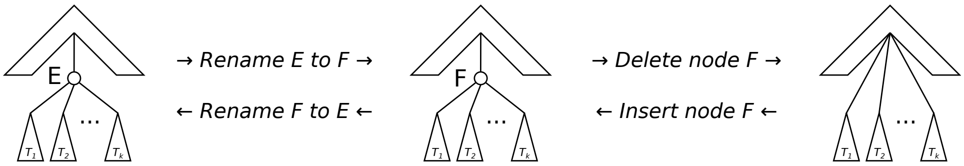


Figure 4.1: Tree Edit Distance operations [Demaine *et al.*, 2009]

Our tool, named DTED (Dependency-based Tree Edit Distance) and published in 2016 [McCaffery and Nederhof, 2016], applies the concept of Tree Edit Distance to the domain of natural language. We have followed the common trend of evaluation metrics of comparing a machine-produced 'hypothesis' translation with a human-produced 'reference' which is assumed to be correct. Both sentences are thus in the same language, and can as a result be processed by identical tools to produce dependency structures which are closely comparable. We have restricted our work to only apply to sentences translated into English, for the varied reasons described in Section 3.3.1, but we do not place any limitations on the 'source' language from which both translations are produced.

4.2 Implementation

We have implemented a prototype of DTED in Python. It requires five separate inputs, representing the sentences to be compared and the tools to be used in that comparison: 1)

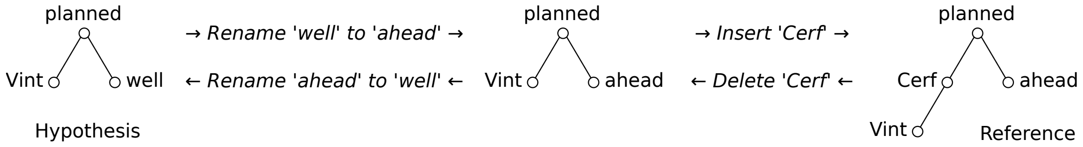


Figure 4.2: The two Tree Edit operations required to convert between an unlabelled reference and hypothesis dependency tree

a hypothesis translation file, 2) reference translation file, 3) word alignment file, 4) parser and 5) tagger. Of these, the hypothesis and reference sentences and word alignments are expected to be provided to our tools through files preprocessed into a convenient format, while the parser and tagger desired are indicated through flags and called automatically by our tools. The nature and exact format of each input has been discussed in Chapter 3, but we will here briefly summarise the various tools we have chosen.

4.2.1 Tree and alignment generation

The information directly used by our algorithm is twofold. The first necessity is the pair of tree structures produced by dependency parsing of the hypothesis and the reference sentences, explained in Section 3.4.3. Note that while such structures contain semantic dependency labels, these are not part of the tree edit distance algorithm and are ignored. The second input is a set of pairings linking each word in each sentence to zero, one or multiple words in the other: a word alignment, discussed in Section 3.4.5.

While our algorithm requires nothing more than this, the tools we employ to generate both pieces of information themselves require input. Specifically, they make use of tokenised sentences (Section 3.4.1), a part-of-speech tagger (Section 3.4.2), and a pretrained file containing data representing relevant features of the language (Section 2.1.2).

Two parsers, aligners and taggers have been used in practice, with each using pretrained language-specific information, where appropriate, from the same online sources as the tools themselves. We work only with hypothesis and reference translations in English (Section 3.3.1), and with one reference translation per hypothesis (Section 3.3.3).

The tools we use are, as described in Chapter 3: the Europarl v6 Preprocessing Tools [Koehn and Schroeder, 2011] for basic text preparation and tokenisation; GIZA++ [Och and Ney, 2003] and cdec [Dyer *et al.*, 2010] for alignment; NLTK's maximum-entropy tagger [Malecha and Smith, 2010] and that provided with the Stanford Parser v3.6.0 [Klein and Manning, 2003b]; and the Stanford Parser alongside the MaltParser v1.8 [Nivre, 2003].

4.2.2 Tree edit operations

The prime component of the DTED algorithm is the actual calculation of the edit operations required to turn one tree into the other. While numerous edit distance algorithms exist [Pawlik and Augsten, 2011], we have implemented the one put forward by [Demaine *et al.*, 2009] due to its relative simplicity and worst-case optimality. In their version of the algorithm, the 'insert' operation, whereby a node is created in one tree X to correspond to a node in tree Y, is represented by a 'delete' of the corresponding node in tree Y.

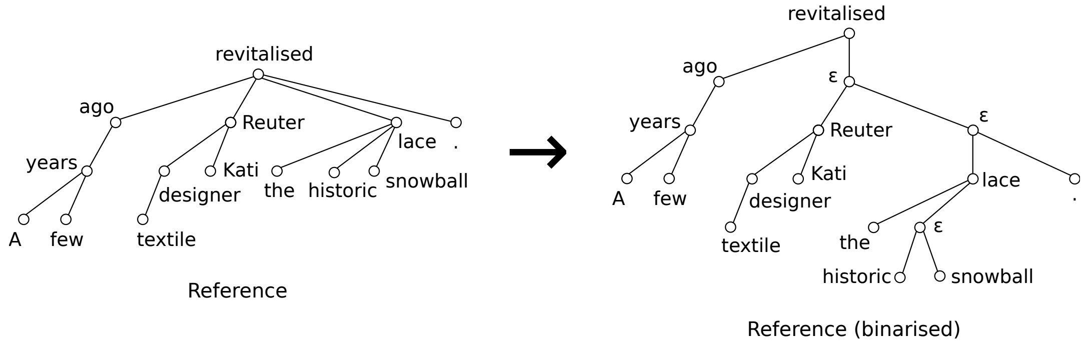


Figure 4.3: A dependency tree as seen by DTED both before binarisation (left) and after (right). Epsilon nodes are indicated by ε.
Reference: A few years ago textile designer Kati Reuter revitalised the historic snowball lace.

The algorithm we use applies specifically to binary input trees. Given that a huge number of our input trees will of course not be in this format, we perform a preprocessing step to binarise the input. This involves creating special ‘epsilon nodes’, which do not represent words in the trees: their use is purely structural. In a recursive procedure, any node with more than two children will have all but the first moved to become children of a newly created epsilon node, which is itself then inserted as the original node’s second child. This is demonstrated in Figure 4.3 on a reference translation included as part of the sentences submitted to the Conference in Machine Translation (WMT2016) [Bojar *et al.*, 2016a]. While calculating tree edit operations, deletion of epsilon nodes is assigned a cost of zero while renaming them is considered prohibitively expensive.

Some configurations of the algorithm (see Section 4.3) separate individual operations into those applied to aligned nodes and those applied to unaligned nodes. The former are assigned a higher priority than the latter: the algorithm strictly minimises the edit count for aligned operations before considering unaligned ones. This separation of priorities focuses on actions which most represent the structural information we have available.

Mechanically, this is done by one type of action being assigned a much greater cost than the other: a single operation of one type may be assigned a cost orders of magnitude greater than one of the other. For example, if preferred actions are assigned a cost per action of 1000 relative to 1 for non-preferred ones, the algorithm will select a solution involving 5 prioritised actions and 11 non-preferred ones – at a cost of 5011 – rather than a solution of 6 prioritised actions but zero non-prioritised ones, costing 6000. The two components – 5 and 11 or 6 and 0 in these examples – can then be trivially extracted once the edit operations have been finalised, and are included in later calculations without the 1000-fold disparity in magnitude.

4.2.3 Normalisation

The initial result of DTED is a count of the number of modifications which must be made to one of the two input trees in order to produce the other. This is already a useful number, but has one significant downside in practice: it does not take into account sentence length. This prevents it from having a clear and simple meaning, consequently preventing its use when comparing multiple sentences. A modification count of 3, for example, could be an

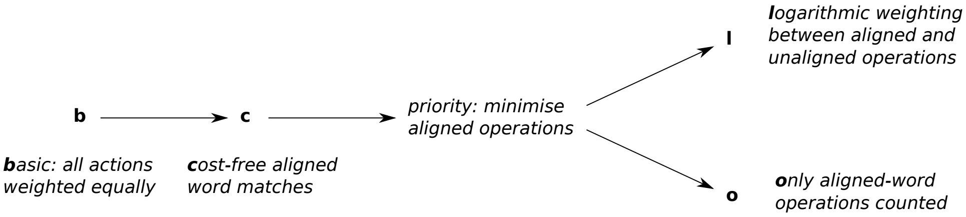

b
basic: all actions weighted equally
c
cost-free aligned word matches
priority: minimise aligned operations
l
logarithmic weighting between aligned and unaligned operations
o
only aligned-word operations counted

Figure 4.4: Summary of the four main flags which can be applied to DTED

excellent score for a pair of sentences each composed of 30 words, while the same score for a pair of three-word sentences is in fact the worst possible.

To mitigate this problem, we have mathematically compared the count of edit operations with the number of words in the two sentences which could be affected. The exact formula for this comparison varies depending on the exact configuration of the tool, as described below. In all cases, this produces a normalised score, a decimal value between 0 and 1, which can be compared meaningfully between sentences.

4.3 Configurations

We have implemented four configurations of DTED, each taking into account in different ways words which do or do not have aligned counterparts in the other tree. Furthermore, each of these configurations can have an additional flag applied which represents a further preprocessing step, flattening the input trees. The different primary behaviours, each assigned a single-letter flag as an identifier, are described in this section, while the universally applicable flattening step is discussed in Section 4.4. Section 4.5 contains a detailed practical example of how each configuration works relative to the others.

The four main configurations are as shown in Figure 4.4, and can be briefly summarised as follows. The most basic configuration of DTED, ‘b’ (Section 4.3.1), represents as closely as possible the tree edit algorithm in its original application-independent form: all actions are assigned equal priority. The ‘c’ flag introduces a caveat to this, with aligned nodes being renameable to each other for zero cost. The ‘l’ flag indicates that operations between aligned and unaligned words are given a relative weighting, while the ‘o’ flag simply ignores all operations on unaligned words; both are discussed in Section 4.3.2.

Each configuration has been run on two types of input. We have run each on normal dependency trees, leveraging the full structural information available. The ‘f’ flag (Section 4.4, continuing Section 3.4.4), refers to the algorithm being instead run on flattened trees: those from which the structural information has been removed.

4.3.1 Traditional edit costs: ‘b’ and ‘c’ flags

The most straightforward way of executing a tree edit distance is to give equal weighting to all operations on all nodes. This is how the algorithm traditionally works in other fields, and how its cousins Word Error Rate and Levenshtein Distances function. This gives us a measure of the structural similarity of the two trees: two identical trees will have the minimum cost, while any sub-optimally placed nodes will need to be deleted at a cost of one action per tree.

We encounter an interesting edge case when two sentences contain exactly the same word. While in other domains two nodes may be considered identical if they have the same value, and thus require no edit operation, we do not consider this to be automatically valid behaviour in our area. This is because two occurrences of the same word may arise from different contexts: for example, an adjective may appear twice to describe two separate nouns within a single sentence pair.

In this situation, the only free operation for each appearance of the adjective in one sentence should arguably be with the appearance in the other sentence relating to the same noun. Happily, we do have a means of determining which occurrences of words match between sentences: word alignment. We are thus able to build this in, and introduce the 'c' flag. This allows aligned words to be matched for no cost using a special free version of the 'rename' operation.

To evaluate the effectiveness of this feature, we have also run DTED without it. By default, 'b'-flag DTED applies a cost to any and all Rename operations, even those with alignments or identical words. This makes 'b' DTED both naïve and strict, with its utility being mainly to observe the improvement produced through the introduction of zero-cost matches in 'c' DTED rather than being itself a good evaluator.

The final score for a given sentence is straightforward to calculate. To produce a score in the range [0,1], we must divide the number of edit operations $dist$ by the maximum possible operation count. In our case, this will involve every single node requiring an edit operation. For comparison with other tools and with intuition, we would prefer a higher score to represent a higher-quality sentence. As a higher $dist$ is indicative of a lower-quality sentence, we invert the score by subtracting it from 1. Given n_H nodes in the hypothesis tree and n_R nodes in the reference, this gives us the formula:

$$score_{config} = 1 - \frac{dist}{n_H + n_R}, config \in \{b, c\} \tag{4.1}$$

Note that without the 'c' flag, the number of edit operations has a minimum of one per node in the larger of the two trees. This is because every single node – in either tree – must have some operation performed on it, either to match it to another node or to remove it. While 'match' operations affect two nodes at the cost of one operation, 'delete' actions affect just one node per operation. The fewest possible operations thus consist of one 'match' operation per node in either tree, with no 'delete' operations being required through mismatches in structure or length.

This thus leads to a best case where $dist = n_H = n_R$, leading to a final score from Equation 4.1 of 0.5. Given that our evaluation (Chapter 7) is based on correlation rather than examination of absolute score values, this difference in range is not expected to have any inherent negative effects on the performance of 'b'-flag DTED.

4.3.2 Prioritising aligned operations: 'o' and 'l' flags

While the most obvious and comparable way of implementing tree edit distance to machine translation is to assign all operations equal weighting, it is far from the only one. In our area, we consider that words which are aligned between the trees can be given a

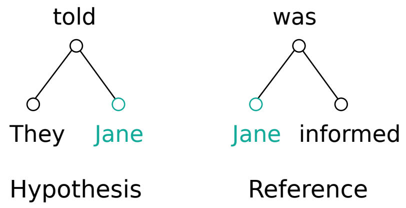


Figure 4.5: Sample parsed dependency trees. Matching colours indicate alignment between nodes; black nodes are unaligned.
Hypothesis: They told Jane
Reference: Jane was informed

much greater measure of confidence than their unaligned fellows. We have thus applied a new restriction to the algorithm: that of preferring to minimise these operations at any cost. Thus, as described in Section 4.2.2, the result of the algorithm will strictly involve moving or deleting as few aligned nodes as possible, with operations on unaligned nodes being considered only as long as the minimal aligned operation count is maintained.

It should be noted that we always combine this functionality with the 'c' flag above, causing aligned nodes to be matched to their aligned counterparts for free. Given that our basic assumption is that aligned nodes are more information-rich than unaligned ones, we need to use the information they provide as completely as possible.

However, using 'b'-flag DTED would provide no way to differentiate matching aligned nodes together by alignment pairs or by their relative positions in their respective trees. This to a significant extent ignores the information contained in the alignments, and prevents a simple intuitive understanding of what the aligned operation count represents.

On the other hand, with the 'c' flag that meaning is clear: the operation count indicates the minimum number of aligned nodes which could not be matched to any counterpart. Put another way, it represents the degree to which the structures of the trees cannot be reconciled: the amount of information encoded in the alignment which cannot be included in the 'best' possible set of matches.

To illustrate the difference in behaviour indicated by both 'o' and 'l' flags, consider the sentences in Figure 4.5. DTED with either the 'b' or 'c' flags activated alone would be unable to match together the aligned word in this sentence pair. Both configurations would produce a cost of 3, resulting from matching the three pairs *They/Jane*, *told/was* and *Jane/informed*. A strict prioritisation of aligned nodes would, however, force the two aligned words to be matched together for an aligned cost of 0, while incurring an increased unaligned-node cost of 4 as all other nodes in both trees would need to be deleted to allow such a match.

This strict prioritisation of operations on aligned nodes poses a new question. In the previous configurations of DTED we considered all non-free edit operations to contribute equally to the final score. However, our strict prioritisation of aligned operations could often lead to very large edit counts for operations on unaligned words. Such an inflated operation count should arguably not lead directly to the conclusion that the sentence is significantly flawed. In fact, the majority or even the entirety of the meaningful information in the score comes now from the aligned words. We must consider how to interpret and normalise the resulting edit score in light of this.

‘o’ flag: aligned operations only

The ‘o’ flag with DTED answers this question very simply, by entirely ignoring all edit operations on unaligned nodes. The assumption behind this flag is that unaligned words represent algorithmic noise whose relevance and quality cannot be determined using the information available in our trees. With no information about the ‘correct’ structure surrounding any of these nodes, or their position within such a structure, such nodes do not contribute any meaningful information and should, for clarity, be omitted entirely.

However, this may not be the case. While we can only confidently and meaningfully interpret operations on aligned nodes, unaligned nodes are not entirely worthless to us. The main indication they provide is this: the more unaligned nodes which can be Matched with a corresponding node in the other tree, as opposed to simply Deleted, the more similar we can consider the two structures to be. This proportion is directly encoded in the matching operation count, as Delete operations incur a cost of 1 per node affected while Rename operations affect two nodes for the same cost and are thus proportionally cheaper. While we cannot know if such matching is between words with similar semantic roles, we make the assumption that sentence trees with similar structures are likely to be semantically related. Despite being formally unsupported, we feel that this assumption is plausible.

It is strengthened when we consider that unaligned Match operations must be compatible with any aligned words which do exist in the trees. Should a large number of words be aligned, a large proportion of unaligned Match operations can be hypothesised to represent any information which was missed by the alignment tool. On the other hand, should many unaligned words need to be deleted, we can conjecture that the aligned words which exist are in dramatically different contexts within their respective sentences. In the opposite situation, when few words are aligned, the unaligned words are ever more important. While our knowledge is weaker without the existence of unaligned nodes to provide context, information relating to unaligned nodes is the only information we have access to with any meaning.

‘l’ flag: logarithmic weighting

We thus calculate a weighting factor for unaligned nodes, allowing us to consider the factors from both unaligned and aligned words. This is based on the proportion of words which are aligned across both trees, exactly as discussed in Section 3.5.1, but in addition to this we have applied a logarithmic element. This is because even with the above arguments we consider aligned words to be the most important factor in determining word order quality, so we disproportionately favour scoring relating to them as long as such information is available.

In practice, we have assigned the logarithmic base 0.1, which results in a multiplier of 0.5 when 30.1% of words in the sentence are aligned, and 0.178 when 75% of words are aligned. It is capped at 0.9 to ensure that aligned words are always considered in the equation. The exact base used could be further refined through empirical parameter-estimation techniques, but for our primarily-exploratory study we did not consider the benefits of such training to be worthwhile. Notably, given the number of sentences available for our experiments, and the non-trivial time required to parse and score each sentence, we preferred to run our tools on more sentences rather than perform many iterations on a much smaller number.

Of course, score components relating to aligned and unaligned nodes can be combined using methods other than logarithmic weighting. Initial experiments were run using a proportional weighting system, with $prop$ below being used in place of log in equation 4.5. However, preliminary results comparing logarithmic and proportional weighting indicated that the former correlated more closely with human judgments, so for simplicity the latter were omitted from the larger-scale experiments we conducted.

We can thus produce simple formulae to calculate the final score for DTED. With the ‘o’ flag (equation 4.2 below), this is the number of edit operations $dist_a$ required by the algorithm on aligned nodes, ignoring those on unaligned nodes, compared with the maximum possible number of such actions which is equal to the total number of such nodes across both trees: a_H and a_R respectively. This is subtracted from 1 to produce a score where a higher number is better, in the same manner as Equation 4.1 above.

With the ‘l’ flag enabled, we must consider in Equation 4.5 the proportion of aligned nodes in the same manner, but also that of unaligned nodes. We thus produce a fraction relating to the number of edit operations $dist_{na}$ required on unaligned nodes relative to the total number of unaligned nodes in the two trees, na_H and na_R respectively.

These are then added together using a weighting, log, which is calculated based on the coefficients described above: our chosen base of 0.1 is raised to the power of the proportion $prop$ of nodes which are aligned across both trees (equivalent to Equation 3.1, page 34). This mathematically cannot be less than 0.1, and is capped at 0.9 should it exceed that value. Exceptionally, if no nodes in the tree are aligned we rely exclusively on unaligned nodes for our score, with a forced log value of 1. Finally, the aligned and unaligned components of the score are subtracted from 1.

$$score_o = 1 - \frac{dist_a}{a_H + a_R} \tag{4.2}$$

$$prop = \frac{a_H + a_R}{n_H + n_R} \tag{4.3}$$

$$log = \begin{cases} 1 & \text{if } prop = 0 \\ min(0.9, 0.1^{prop}) & \text{if } prop > 0 \end{cases} \tag{4.4}$$

$$score_l = 1 - \left((1 - log) \times \frac{dist_a}{a_H + a_R} + log \times \frac{dist_{na}}{na_H + na_R} \right) \tag{4.5}$$

Note that if either no word or every single word is aligned across both trees, several of these formulae become mathematically undefined. This is circumvented by treating any fraction across all equations for which the bottom line equals zero as if the entire fraction had a flat value of zero. For example, if no words are aligned then the final score using the ‘o’ flag will be 1, while the score using ‘l’ will be solely based on unaligned nodes.

4.4 Tree flattening: ‘f’ flag

Our creation of DTED has been led by a number of different fundamental priorities, described in Section 3.1 (page 20). Key among these is that any tool we create should permit a general evaluation of the effect of structure on the word order evaluation process (Section 3.2.2).

It is in the interests of fulfilling this and other goals that we have produced simpler versions of DTED, relying on the 'flattened' structures produced through the transparent preprocessing step described in Section 3.4.4. These variants are not intended to leverage the full structural information we use normally, but instead to produce a meaningful, structure-free baseline for comparison.

4.4.1 Merit & expectations

The process of flattening has two important sets of merits: one intrinsic and one relative. The latter is described in more detail in Section 3.4.4: using flattened input to purely evaluate the effect of structure on our tools. By running identical algorithms on input which either does or does not include genuine structure, we allow a clear and fair observation of the effect of this feature in isolation, eliminating all intrinsic effects and assumptions made by the algorithm itself.

The intrinsic merits of our flattened systems vary according to the system itself, and depend on what it is in fact measuring. The flattening of the input trees represents a severe change to DTED, altering its most fundamental assumption: that the structure of the input trees with which it is provided inherently contains merit. Without this assumption, we are forced to reconsider in detail the question of what information is encoded in the scores from each configuration.

'fb' DTED

The basic configuration of DTED, 'b', is intended to provide a numericisation of the difference between two structures. However, when the two structures are guaranteed to both be similar, we must consider what factors remain to be taken into account. These are twofold: the number of words in each sentence, and the alignment of those words. The latter of these factors is only considered when the 'c' flag is applied, allowing the alignment to treat aligned words differently from unaligned ones.

Unfortunately, without this flag the algorithm has extremely little to rely on: it will simply match all nodes by position, then delete any remaining words in the longer of the two trees. This is illustrated in Table 4.1 (page 52). Note that our normalisation formulae (Section 4.3) rely on the lengths of the sentence, with the consequence that the only variation which can be observed in these scores is between the number of nodes matched – which must always equal the length of the shorter sentence – and the number deleted. This is effectively a complex encoding of the difference in lengths between the two sentences, leaving 'fb' DTED little practical merit in evaluation of real quality.

'fc' DTED

With the addition of the 'c' flag, flattened DTED is able to incorporate more meaningful information, namely word alignment. By directly mapping aligned words between the trees for free, the algorithm can produce an edit distance reduced by any number up to the count of pairs of aligned nodes. In this way, it approximates the 'metric' of observing the percentage of words which are aligned in the sentence (Section 3.5.1). This is arguably a useful measure of quality, one which is investigated empirically in Chapter 7.

However, flattened ‘c’ DTED takes into account more than just the number of words which are aligned: it also takes into account their relative positions in the sentence. This is relevant in situations where hypothesis and reference use (or are close to using) exactly the same structure, but as with non-flattened DTED it becomes inadequate in more complex situations.

Consider sentence pair 5 in Table 3.1, where the only arguable ‘errors’ are a mismatch in structure. With structural information deliberately removed from the sentence we would expect it to score highly, as a result of a high proportion of aligned nodes which we can infer from the identical word choice. However, flattened ‘c’ involves a certain level of structural inference based on the surface order of words, and will use this to assign this sentence pair a low score.

Given the above two perspectives, flattened ‘c’ DTED represents a middle ground between a structure-independent evaluation technique – namely observing the proportion of words which can be aligned – and a primitive structured one represented by the similarity of their ordering.

‘fco’ and ‘fcl’ DTED

While most of the merit of ‘fc’ DTED comes from a relatively complex observation of aligned nodes, the ‘o’ flag provides a more simplified inspection: it considers only those aligned nodes which must be mapped to nodes other than their aligned partners. In a flattened tree, this takes on a very specific meaning: counting the number of aligned words which are in a different order relative to each other in the two trees.

This is a much more basic evaluation criterion than the structural one for which the algorithm was designed, but nonetheless represents something potentially useful. Indeed, counting the number of mismatches between individual words’ orders is the basis of numerous existing systems, as discussed in Section 2.3.1. The specific addition of ‘fco’ DTED is the attempt to actively minimise the number of ‘imperfect’ operations, as opposed to merely observing all mismatches.

In practice, this results in a similar task to that of Kendall’s τ, with that of DTED being the simpler of the two in that any nodes considered incorrectly placed are immediately deleted (or suboptimally matched) in up to one operation each. The impact of such errors is thus not scaled according to the number of nodes relative to which the deleted node is incorrectly positioned.

To illustrate this, recall the example of Kendall’s τ in Chapter 2 (Table 3.2, page 34). In it, we can see that with just two erroneous words in the sentence being inspected, Kendall’s τ considers them to be collectively part of 7 ‘discordant pairs’, resulting in a correlation value under 5%. On the other hand, were a flattened version of DTED to inspect the same sentence pair, it would be unable to do more than delete both incorrectly positioned words, which could be considered underestimating the real severity of the two words’ misplacement.

This simplification may not be a positive one, as this discarded information represents the severity of the error of the affected words, thus providing additional detail which may be relevant to humans’ judgment on order quality. We consider ‘fco’ DTED to be worthy

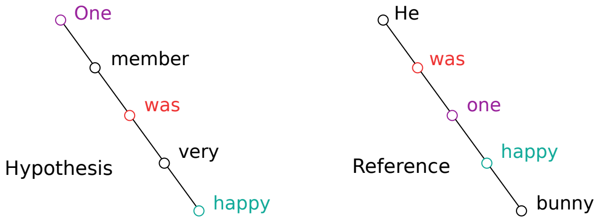


Figure 4.6: Sample flattened dependency trees, using a similar visual convention to Figure 4.5
Hypothesis: One rabbit was very happy
Reference: He was one happy bunny

of investigation for this reason: by considering its ability to predict human judgments on word order alongside that of Kendall's τ, we can deepen our understanding of the importance of this aspect of error severity.

The 'l' flag introduces a new type of information: a limited representation of the similarities of unaligned nodes between the trees. With flattened structures, this primarily involves comparing the quantity of such nodes similarly to 'fb' DTED, though the 'o' flag alters this behaviour to be very specific: it inspects the spacing – measured in unaligned nodes – between pairs of aligned words which can be optimally matched together.

In effect, the absolute priority of minimising aligned node movements produces a series of points which cannot be altered based on considerations relating to unaligned nodes. We can thus conceptually split the sentence into pairs of substrings, delimited by the matched aligned nodes. Each of these is then processed according to the same rules used by 'fb' DTED, with the only variation being based on the difference in their lengths. In this way, the unaligned edit count represents an aggregate of the difference in distribution of unaligned spaces between aligned pairings.

For example, in Figure 4.6 the algorithm will be able to match the two pairs *was/was* and *happy/happy*, but will be unable to match *One/one*. This leaves three disparate sections of each sentence: before *was*, between *was* and *happy* and after *happy*. Each of these sections must be processed separately, as the algorithm is unable to affect multiple segments at once – e.g. matching *One* from the first block in the hypothesis to *one* in the second in the reference – without destroying a zero-cost aligned node pairing (*was/was* here) in the process.

The actual relevance of this segmentation to human evaluation of ordering is not clear, but we can hypothesise that sentences with large quantities of paired words but very different numbers of unrelated words between them may not be of high quality. This factor is arguably not related directly to word ordering, but is more relevant to overall holistic sentence quality. In any case, we feel it is worthwhile to investigate empirically, as with 'fo' DTED, to observe the real-world relevance of these effects.

4.5 Example

To fully understand the operation of DTED, it may help to observe it in practice. We have thus run each different configuration of the algorithm on a specific sentence, and

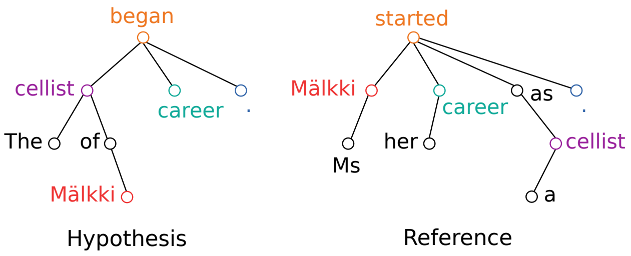


Figure 4.7: Sample parsed dependency trees, using a similar visual convention to Figure 4.5
Hypothesis: The cellist of Mälkki began career.
Reference: Ms Mälkki started her career as a cellist.

noted both the edit operations generated and the mathematical operations done on these to produce final scores.

The sentence we have chosen is taken from the WMT 2015 corpus [Bojar *et al.*, 2015], with the source in Finnish and the hypothesis translation produced by the University of Sheffield's stemmed system [Smith *et al.*, 2015].

Figure 4.7 shows dependency trees for the two sentences, with aligned words highlighted in matching colours. All pairs of words shared by both sentences are aligned, as are 'started' and 'began'. Given these trees, the operations DTED performs on the trees vary significantly based on the active flags.

The operations performed by DTED when using the 'b' flags are indicated in Table 4.1. Without the behaviours introduced by other flags, the algorithm matches together what nodes it can, assuming no labels are equal, producing six Rename operations. Four Delete operations are required on nodes which simply represent disparities in the structure of the trees, resulting in a total matching $dist$ of 10.

When the trees are flattened through the 'f' flag, the operations performed by 'b' DTED change. In practice, the sets of nodes which are matched together are very different, but the overall number of modifications is only slightly altered. Specifically, as shown also in Table 4.1, the lack of structure allows one more 'match' operation to be performed, replacing two 'delete' operations and thus producing an overall $dist$ of 9 instead of 10.

With the 'c' flag enabled, many operations become free of cost, which significantly alters the actions selected as per Table 4.2. In taking advantage of these opportunities to reduce cost by pairing together as many aligned nodes as possible, the algorithm is forced to match fewer unaligned nodes together: only two are matched which are not aligned. However, four 'match' operations between unaligned nodes are available at zero cost, resulting in a cheaper overall solution ($dist = 7$) despite the addition of two more 'delete' actions compared to the 'b' solution. Note that the aligned nodes *cellist/cellist* could not be matched together in an optimal solution.

The additional priority placed on aligned-node operations when applying the 'o' and 'l' flags does not in this case lead to any cheaper solution than that of 'c' DTED. However, the operations performed on aligned nodes are now considered separately, resulting in $dist_a = 2$ aligned nodes deleted out of $a_H = a_R = 5$ in the hypothesis and reference trees respectively, and $dist_{na} = 5$ 'rename' and 'delete' operations performed on the $na_H = 3$

DTED with ‘b’ flag			
	Hypothesis	Reference	Cost
Matched	began	started	1
	The	Mälkki	1
	of	career	1
	Mälkki	her	1
	.	.	1
	career	as	1
Deleted	cellist		1
		Ms	1
		cellist	1
		a	1

DTED with ‘fb’ flags			
	Hypothesis	Reference	Cost
Matched	The	Ms	1
	cellist	Mälkki	1
	of	started	1
	Mälkki	her	1
	began	career	1
	career	as	1
	.	a	1
Deleted		cellist	1
		.	1

Table 4.1: Edit operations calculated by ‘b’ DTED for original and flattened sentences shown in Figure 4.7

DTED with ‘c’ flag			
	Hypothesis	Reference	Cost
Matched	The	Ms	1
	Mälkki	Mälkki	0
	began	started	0
	career	career	0
	.	.	0
Deleted	cellist		1
	of		1
		her	1
		as	1
		a	1
		cellist	1

DTED with either ‘co’ or ‘cl’ flags			
	Hypothesis	Reference	Cost
Matched	The	Ms	1
	Mälkki	Mälkki	0
	began	started	0
	career	career	0
	.	.	0
Deleted	cellist		1000
	of		1
		her	1
		as	1
		a	1
		cellist	1000

Table 4.2: Edit operations calculated by ‘c’, ‘co’ and ‘cl’ DTED for sentences shown in Figure 4.7. Note that flattening does not in this case alter these actions.

and $na_R = 5$ unaligned nodes in the hypothesis and reference trees respectively.

It is interesting to note that with this sentence specifically, flattening the trees before running the algorithm does not produce different sets of edit operations for the ‘c’, ‘co’ or ‘cl’ configurations. This is a consequence of the two trees having similarly ordered aligned nodes irrespective of structure. Specifically, when considering the dependency structures ‘began’/‘started’ is the root word in both examples, with ‘Mälkki’, ‘career’ and ‘.’ being within the first, second and third subtrees beneath that node in both cases while ‘cellist’ does not occur in the same subtree.

In the flattened tree, we can more simply observe that ‘Mälkki’ is before ‘began’/‘started’ in both cases, which is in turn before ‘career’ then ‘.’, with the order of ‘cellist’ lacking a similar convenient shared descriptor. Were the two trees more different in structure (or at least the structure relating to the aligned nodes), we may have observed more noticeable effects when flattening the trees.

With the hypothesis tree containing $n_H = 7$ nodes and the reference containing $n_R = 9$, we can now normalise each of the calculated operation counts according to the equations in Section 4.3, producing final scores as follows:

$$score_b = 1 - \frac{10}{7+9} \approx 0.375 \tag{4.6}$$

$$score_{fb} = 1 - \frac{9}{7+9} \approx 0.438 \tag{4.7}$$

$$score_{fc} = score_c = 1 - \frac{7}{7+9} \approx 0.563 \tag{4.8}$$

$$score_{fo} = score_o = 1 - \frac{7}{5+5} = 0.700 \tag{4.9}$$

$$prop = \frac{5+5}{7+9} \approx 0.625 \tag{4.10}$$

$$log = min(0.9, 0.1^{0.625}) \approx 0.204 \tag{4.11}$$

$$\begin{aligned} score_{fl} = score_l &\approx 1 - \left((1 - 0.204) \times \frac{2}{5+5} + 0.204 \times \frac{5}{2+4} \right) \\ &\approx 1 - (0.159 + 0.170) \approx 0.671 \end{aligned} \tag{4.12}$$

4.6 Publication

Part of the work described in this chapter has been previously published at the First Conference on Machine Translation in 2016 [McCaffery and Nederhof, 2016]. The paper was submitted as a system description to the Metrics Task. Additionally, a set of 160,951 scores was contributed, to be evaluated using the same process as the rest of the systems submitted to the Conference. We also ran a smaller-scale experiment using translations produced during WMT 2015 [Bojar *et al.*, 2015].

For the sake of simplicity and conformity with other submissions, we only submitted scores from a single configuration of DTED to the main Shared Task. In the interests of brevity, we chose the one with the fewest dependencies, namely activating only the ‘b’ flag. The experiments were thus limited in scope, unable to assess the full extent of DTED’s functionality.

In addition to limitations from the configuration submitted, the evaluation technique in the Conference was not fully aligned with the goals of our tool. While DTED has been designed to investigate the ordering of words, the judgments against which it was compared were related to more general quality. While the exact criteria with which to determine this quality were left up to individual judges, they can be assumed to account for word choice, fluency and other factors which DTED is specifically intended to bypass.

The experiment we ran prior to submission, whose results were generated on WMT 2015 data and were reported in the paper, included several minor variations on DTED's configuration. Specifically, we ran the tool on flattened structures (Section 4.4) as well as normal dependency trees, and calculated system-level scores (Section 3.3.3) both using alignment-based weighting and arithmetic means. We found that the weighting had little effect on the scores, while the flattening significantly worsened performance as predicted. Note that the WMT 2015 data included in this experiment was in the same format as that of WMT 2016, resulting in the same mismatch in objectives described above.

Due to the mismatch between the priorities of our tool and those of the WMT evaluations, we do not rely on the published results for our primary evaluation of DTED. Instead, see Chapter 7 for a more tailored and in-depth experimental setup.

4.7 Limitations

While we believe DTED to be a highly relevant tool for investigating structure in machine translation, it is nonetheless imperfect. Its primary limitation comes in its lack of consideration for the severity of the errors it observes.

Specifically, when observing any individual node in a tree, the algorithm can assign one of exactly three judgments: the word is correct (a zero-cost 'match' operation is possible), roughly approximates to something in the opposing tree (a non-zero cost 'match') or does not correspond with anything in the other tree (requiring a 'delete').

These three options do not provide much opportunity for nuance in DTED's scores. Consider again the sentences in Table 3.1 (page 21). Sentence pair 3 may be considered a serious mistake, causing the intended meaning of the sentence to be difficult for a human to extract, while the mismatch in order in sentence pair 2 is arguably not even an error.

While these two facts are likely to be encoded in a dependency parse of the two sentences, in both cases DTED will detect an irreconcilable difference and assign similar penalties. Even in the case of sentence pair 4, where the mismatch is localised to a particular subtree, DTED will be forced to apply at least one maximum-severity 'delete' operation despite the impact on comprehensibility being almost nonexistent.

This lack of flexibility necessarily limits the accuracy with which DTED can encode flaws in translation word order. This in turn makes its relevance to our central question – that of assessing the impact of structure on evaluation – very specific. We can consider DTED to be an assessment of the number of errors which exist in a given sentence: a somewhat naïve broad-strokes investigation of quality.

This approach provides a limited intrinsic insight into the question of structure, along with more comparative insight when considered along with other tools whose goal is to investigate the severity of the mismatches detected in addition to their quantity. This is the goal of our next project, DERP, described in Chapter 5.

CHAPTER FIVE

DERP: DEPENDENCY ERROR RATING WITH PATHS

5.1 Introduction

While our first tool, DTED (Chapter 4), represents a number of subtly different ways to measure the number of inaccuracies to do with word order in machine-produced translations, its approach to solving the problem is far from the only one possible.

DTED has a number of limitations, as described in Section 4.7: most notably these are tied to its ability to do little more than count individual mismatches between words' positions. In practice, we may wish to additionally quantify the severity of those mismatches in order to produce a more nuanced score. DTED also ignores a fundamental feature of dependency parses, namely the labels associated with dependency links.

Our second tool, DERP (Dependency Edit Rating with Paths), represents an attempt to account for the more detailed information we would like, through the comparison of paths between aligned nodes in the two trees. Its processing takes into consideration the dependency labels along those paths. Similarly to DTED, DERP is based on an algorithm which, while well-known in its own area, has to our knowledge hitherto never been applied to the field of machine translation evaluation. This is Kruskal's algorithm for the minimum spanning tree [Kruskal, 1956]. While our final algorithm is similar to the existing one, we must nonetheless adapt this last to apply to our own domain.

Specifically, the input to DERP is broadly the same as that to DTED. Given a hypothesis sentence and a reference sentence, we parse both using one of two off-the-shelf parsers to produce dependency trees. Separate alignment tools also produce symmetric mappings between words in the two sentences. These three pieces of information are then passed to the algorithm, which must finally produce a measure of their similarity in the form of a normalised score in the range [0,1].

In order to address the above shortcomings of DTED, DERP has a very specific remit: evaluating word order through detecting not only *which* nodes are incorrectly positioned, but also *how severe* such mistakes are. Our primary assumption is that through information about all sentence errors which is more detailed than that used by DTED, DERP will be able to provide more reliable and accurate indications of the quality of the word ordering in any given sentence.

While this can be done without the use of sentence structure, our secondary hypothesis is that by building in as much syntactic information as possible in the form of dependency trees, we can provide more relevant information for both our tools and thus increase their accuracy. This contributes to our overarching goal of investigating the effect of dependency structures on this area of evaluation. To this end, we have provided two versions of DERP: one which utilises parsed dependency trees, and another which eliminates the structural information from such a parse before its execution as described in Section 5.10.

5.2 Motivation & definitions

To pursue our first hypothesis, namely utilising more detailed inspection of our sentences than that of DTED, our first task is to determine which features of the trees we inspect might contain information about the severity of any given word order mismatch. The method we have chosen focuses on the paths between different nodes. Our central assumption is that a comparison of the path between two nodes in a hypothesis tree against the corresponding path between two related nodes in a ‘correct’ reference tree will provide a measure of information about whether those two nodes are correctly positioned relative to each other.

5.2.1 Dependency trees

Before we can discuss the encoding and comparison of paths, we must introduce certain basic concepts. First and foremost among these are dependency trees, defined in more detail in Section 3.4.3. We recall that a *dependency tree* consists of a series of nodes representing words, linked together by edges whose labels indicate syntactic relations between those words.

The *nodes* or *vertices* within these trees represent individual words, while *edges* refer to the dependency relations between nodes. Three pieces of information are associated with each edge: the two nodes it connects, and the dependency *label* associated with the relation. The label encodes in a simple manner a meaningful measure of the semantic nature of the relationship between the two words. We will often consider edges as members of *path*s between pairs of nodes: such paths can pass through any number of adjacent edges.

We use a variable name such as n_D to refer to a node in a dependency tree D. The set of nodes in such a dependency tree is denoted by N_D. Paths between nodes are referenced by a 2-tuple (n_D, m_D) indicating the nodes they connect. We do not use a separate notation to refer to individual edges, which are simply considered to be paths of length one.

Dependency trees are always considered in pairs, with one being referred to as D and the other as D'. Generally, the two trees refer to a machine-produced hypothesis translation and a human-produced reference translation assumed to be correct, though which of these is referred to as D and which as D' is unimportant. Despite this, purely for consistency we refer to reference trees as D and hypothesis trees as D' in the figures throughout this chapter.

Pairs of dependency trees have one important link, as discussed in earlier chapters: individual nodes may be *aligned* through a separate mapping. $C \subseteq N_D \times N_{D'}$ is a relation between nodes indicating alignment. We say $n_D \in N_D$ and $n_{D'} \in N_{D'}$ are *aligned* if $C(n_D, n_{D'})$, with each node being a *counterpart* of the other, and define A_D and $A_{D'}$ to refer to the sets of all aligned nodes in the two trees.

$$A_D = \{n_D \in N_D \mid \exists n_{D'} \in N_{D'} : C(n_D, n_{D'})\} \tag{5.1}$$

$$A_{D'} = \{n_{D'} \in N_{D'} \mid \exists n_D \in N_D : C(n_D, n_{D'})\} \tag{5.2}$$

A key assumption we make, in keeping with a common trend in our field described in Section 3.4.5, is that while any number of nodes in D' may be aligned to a single node in D, the inverse is not true: no node in D' may be aligned to more than one node in D. It is still unimportant which of the two is the hypothesis and which the reference. This assumption of one-to-many ordinality is a simplification for the sake of clarity, and can be partially relaxed without loss of generality as discussed in Section 5.7.

Alignment mappings permit us to relate pairs of paths in addition to pairs of nodes: a *path pair* $(n_D, m_D, n_{D'}, m_{D'})$ refers to two paths (n_D, m_D) and $(n_{D'}, m_{D'})$ such that $C(n_D, n_{D'})$ and $C(m_D, m_{D'})$.

5.3 Paths

To evaluate in a granular manner the differences between trees, we investigate individual paths between pairs of nodes. Our key assumption is that in two trees with similar structures, the paths between pairs of aligned nodes will have similar features.

To assess the quality of an individual path in one dependency tree relative to an equivalent path in another, we apply a series of edit operations on the relevant features, calculated using the well-known technique of Levenshtein distances [Levenshtein, 1965], also an inspiration for DTED. However, in order to calculate Levenshtein distances between paths, we must first encode them into comparable strings.

5.3.1 Path encoding

When encoding paths into a format with which we can calculate disparities, we must consider which features will provide meaningful information. We consider three aspects of a path to be important for any comparison: the horizontal direction (left to right or right to left); the vertical direction (which edges are ascending and which descending); and the actual dependency labels of those edges.

To encode the horizontal direction of the path, introduce two *direction symbols* indicating the relative positions of the first node n and last node m of the path. Direction here is defined in terms of the original sentences, such that two direction symbols are possible:

α: the word corresponding to n is to the **left** of that of m

β: the word corresponding to n is to the **right** of that of m

To account for the vertical direction of the edges in any path, we separate it into two components. Each is an ordered sequence of dependency labels, representing the edges respectively before and after the 'highest' node p: the node with least depth from the root. Note that in the special cases when either n or m is in fact the highest node, one or other of these components will be empty.

We can now encode individual paths in a triple containing all relevant information: its three members refer to the string χ of labels from the *ascending* portion of the path (from n to p), the direction symbol ψ alone, and the string ω of labels from the *descending* portion of the path (from p to m). We refer to these triples as $encoding(n, m) = (\chi, \psi, \omega)$. We can for example represent the two paths between nodes *the* and *Sherlock* in the hypothesis and reference sentences in Figure 5.1 as the triples $(\text{DET DOBJ}, \beta, \text{NSUBJ})$ and $(\text{DET NSUBJ}, \alpha, \text{DOBJ CMP})$ respectively.

5.3.2 Path disparity

Having encoded in strings the features of paths which we wish to consider, we must apply our chosen comparison technique of Levenshtein distances [Levenshtein, 1965] to produce a measure of the disparity between them. We refer to Levenshtein distances through the function l, which takes two strings and returns a count of the operations required to convert one to another.

In our implementation of l, we permit substitution (rename) operations in addition to deletions and insertions, all assigned equal cost. This is to allow for words occurring

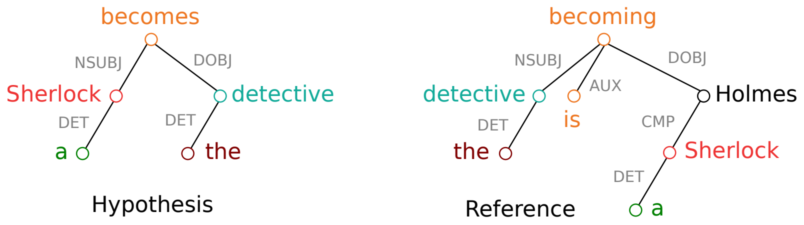


Figure 5.1: Sample dependency trees. Matching colours indicate aligned words, while nodes in black are unaligned. Grey capital letters indicate dependency labels.
Hypothesis: a Sherlock becomes the detective
Reference: the detective is becoming a Sherlock Holmes

in a similar position in a tree but with different labels to be matched at a reduced cost: while all operations are equally costly in themselves, substitutions reconcile two disparate nodes in a single operation, providing a lower cost per affected node than a combination of deletions and insertions. We consider that trees with a similar structure but disparate labels may have been subject to relatively minor parse errors, e.g. simple mis-tagging, or a dramatically different set of tags being required due to the insertion of one erroneous word.

However, we do not permit transpositions (swaps) as unit operations. This is for the simple reason that as we are investigating word ordering, we wish to assign a maximal penalty – namely two deletions and a substitution – for a pair of transposed labels A B and B A: i.e. to any error which is simply a representation of mismatched ordering.

This is strengthened by the observation that transpositions in dependency labels are rarer and less intuitively simple than those of individual words. This is a result of the fact that the order of dependency edges is tied to the ancestry of the relevant nodes: two swapped labels indicate that the relationships between a node and its parent, and the same node and one of its children, have been inverted. This has no intuitively obvious interpretation, in contrast to the relatively simple situation where two words have been swapped. The latter case is likely to be a minor error in ordering, such as an adjective and noun being swapped between different qualities of translation from French to English.

For any given pair of paths we make three separate calls to l, calculating the disparity in the ascending and descending portions of the paths and between their direction symbols. The results of these three calculations are added to produce an overall disparity L.

We avoid combining the three features into a single string for each sentence to ensure that the direction symbol is treated entirely separately from the dependency-label characters in the strings. Were the components to be combined for a single Levenshtein calculation, it would be possible to simply delete the direction symbol from each string in two operations, allowing matching of strings in counter-intuitive manners.

To illustrate this, consider a path pair whose paths in D and D' are encoded as $($XCOMP XCOMP$, \alpha,)$ and $(, \beta,$XCOMP XCOMP$)$ respectively. Each of these paths is composed of two edges with identical labels, linking a node to its grandparent in one tree and to a grandchild in the other. Our intuition dictates that this is a significant order error, dramatically altering the relative meaning of the words: it should thus receive as great a penalty as possible.

However, were we to combine all components to calculate l using the two strings XCOMP XCOMP α and β XCOMP XCOMP, it would be possible to simply delete both direction symbols. With the remaining symbols matching without any further operations required, this would result in a cost of 2: much cheaper than the 5 operations required when the paths' components are compared individually.

In addition to the actual Levenshtein distance between paths, we introduce L_{max}, the maximum possible edit distance between two paths. We will use this when normalising the final scores, as discussed in Section 5.8. We observe that such a distance between any two strings involves performing one rename operation to match every symbol in the shorter of the two strings with one in the longer, then deleting any remaining symbols in the longer string. This results in a maximum cost for any given string pair equal to the length of the longer of the two, and a maximum cost for any encoded path pair equal to the sum of the three maxima for each pair of strings in the paths' triples.

For $encoding(n_D, m_D) = (\chi_D, \psi_D, \omega_D)$ and $encoding(n_{D'}, m_{D'}) = (\chi_{D'}, \psi_{D'}, \omega_{D'})$:

$$L((n_D, m_D, n_{D'}, m_{D'})) = l(\chi_D, \chi_{D'}) + l(\psi_D, \psi_{D'}) + l(\omega_D, \omega_{D'}) \tag{5.3}$$

$$L_{max}((n_D, m_D, n_{D'}, m_{D'})) = max(|\chi_D|, |\chi_{D'}|) + 1 + max(|\omega_D|, |\omega_{D'}|) \tag{5.4}$$

Note that $L(n_D, m_D, n_{D'}, m_{D'}) = L(m_D, n_D, m_{D'}, n_{D'})$. This is because distances are defined purely by differences between the paths, rather than by any absolute ordering. Should m_D and n_D be transposed, the requirement that the two end nodes of a path be aligned dictates that $m_{D'}$ and $n_{D'}$ must also be exchanged. Thus, both paths will be encoded in the reverse order, without affecting the essential characteristics of the resulting strings.

5.4 Spanning

While the concepts introduced in the previous section allow us to calculate the disparity between the paths between any pair of nodes in each of two trees, it does not yet give us any information about the overall quality of a tree. Recall our central assumption above: that the comparison of paths between nodes in a hypothesis and a 'correct' reference tree gives us an indication of the quality of the former's nodes' positioning relative to each other. To scale this up to apply to a full tree, we consider it necessary to produce such comparisons for *all* nodes across both trees. An aggregate summary of all such positions can then be expected to represent the level of (dis)similarity between the two entire trees.

Happily, we neither wish nor need to calculate path distances between every possible pairing of nodes. While such calculations would be extremely time-consuming to calculate, they would also have more theoretically important downsides.

Consider the case illustrated in Figure 5.1 (page 59), where the subtrees rooted at *detective* and at *Sherlock* have been swapped. This arguably refers to only two order-related mistakes – the improper positioning of two nouns – but has secondary consequences on the two determiners *the* and *a*. Were we to consider every possible pair of nodes in our sentence-level score, we would necessarily include both $a =$ *(a,becomes,a,becoming)* and $b =$ *(the,becomes,the,becoming)*. Inspection of each of these pairs would detect directionality mismatches between the positions of both determiners.

While this is correct, in that the nodes are in fact wrongly positioned in the two trees, the actual fault arguably lies with (and only with) the positions of their respective parents, 'Sherlock' and 'detective'. These more intuitively relevant errors would be detected when inspecting the pairs $c =$ *(Sherlock,becomes,Sherlock,becoming)* and $d =$ *(detective,becomes, detective,becoming)* respectively.

Including the two path pairs a and b, in addition to c and d, would thus twice take into account the errors relating to the two nouns, as a necessary consequence of the erroneous nodes having dependents. We consider that this would be inappropriate, as in a significant intuitive sense the determiners are in fact correctly positioned: as immediate dependents of the appropriate noun.

Our solution to avoid this duplication is to require that as much as possible, no pair of paths can be considered more than once in our evaluation. In the example above, the paths described by a and b traverse the sections of the trees also covered by c and d. Using this fact to reject a and b prevents any mismatch from being considered in more than one path.

In order to produce a score representing as much information as possible about a dependency tree, we must fulfill another obvious criterion: the pairs of paths we select must, if only indirectly, relate every single aligned node in both trees to every other node in the same tree. Failing to do so would result in the omissions of certain paths which could contain errors, resulting in unexpectedly encouraging scores if the omitted paths did contain errors, or discouraging ones if they did not.

5.4.1 Meta-graphs

As a result of the two constraints above – relating every pair of nodes without considering any path more than once – we introduce the principle of a spanning tree. For this, we must first introduce the concept of a *meta-graph*: an undirected weighted graph distinct from, yet based on, one of the original dependency trees.

Before defining a meta-graph in detail, we must briefly remark that a meta-graph can be generated from either or both of the original dependency trees. However, we will in this section consider only one meta-graph for any given pair of dependency trees: we show in Section 5.5 that we have no need of the meta-graph which could be created from the opposing tree in order to accomplish the goals described above.

A meta-graph is a simplification of the concept of a dependency tree, with a number of important differences separating the two. The first of these is that the nodes in a meta-graph M do not include any unaligned nodes: the set of its nodes N_M is thus equal to A_D, the set of aligned nodes in the dependency tree D on which it is based. The alignment relation on these nodes remains unchanged, as does our assumption of one-to-many alignment ordinality in one direction only. Thus, any meta-graph node n_M must have exactly one counterpart $n_{D'}$ in the dependency tree D' on which M is not based.

The edges between nodes in a meta-graph are linked to those of both related dependency trees, in one specific aspect. We assign the *weight* $|(n, m)|$ of each edge in a meta-graph based on the paths in the dependency tree between the edge's two end-point nodes $n, m \in N_M = A_D$. Such weights are calculated based on the disparity between the path in the dependency tree D on which the meta-graph is based, and the related path in the opposing dependency tree D'. The disparity between the paths is calculated using Levenshtein distances, as described above.

Our assumption of alignment ordinality allows an edge case to arise with path weights: multiple nodes n, m to be aligned to the same counterpart $n_{D'} = m_{D'}$. In this case, the path $(n_{D'}, m_{D'})$ contains no edges, preventing us from producing any meaningful Levenshtein distances. Rather than comparing (n, m) to this empty path, we assign a flat cost of 0 to the edge.

$$|(n_D, m_D)| = L((n_D, m_D, n_{D'}, m_{D'})) \quad \text{for } C(n_D, n_{D'}), C(m_D, m_{D'}), n_{D'} \neq m_{D'} \tag{5.5}$$

$$|(n_D, m_D)| = 0 \quad \text{for } C(n_D, n_{D'}), C(m_D, m_{D'}), n_{D'} = m_{D'} \tag{5.6}$$

Note that meta-graphs are inherently strongly connected. This is a consequence of their edges being based on the nodes in a dependency tree which itself, by nature, is connected. As such, a path must exist between every pair of nodes in the tree. From each of these paths, a direct edge can be generated within the related meta-graph.

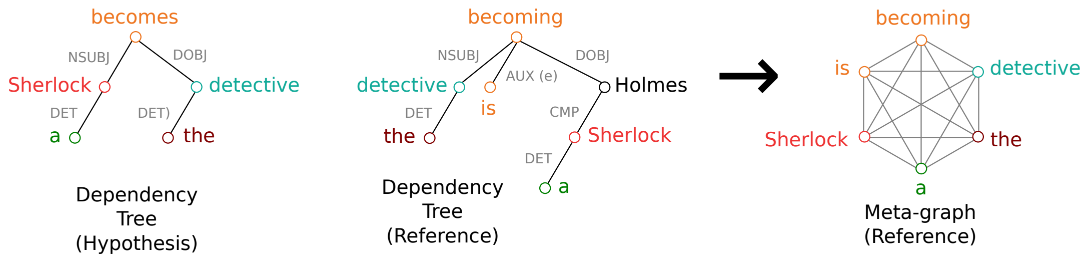


Figure 5.2: Dependency trees from Figure 5.1 with the meta-graph generated from the reference tree, using a similar visual convention to Figure 5.1. Weights for meta-graph edges are described in Table 5.1 (page 69).

Using the formalism of a meta-graph, we can now fulfill the two criteria laid out earlier: collecting path distances while taking into account every aligned node in a given dependency tree with respect to every other from the same tree, and yet inspecting none more than once. We observe that any solution to these constraints will represent a special subgraph of the associated meta-graph: a *spanning tree*.

A spanning tree is a tree which connects every node in a graph: it thus provides a path from any node to any other – thus allowing each node to be taken in the context of every other – with its nature as a tree preventing cycles. A cycle, were one to exist, would represent multiple ways to travel from one node to another, in turn implying the relative positions of the two nodes being considered in more than one way: a duplication of the inspection of the disparity between the two nodes' positions.

5.4.2 Minimality

We have thus far determined that in order to evaluate the severity of the mismatches in position between aligned nodes in a dependency tree, we can produce spanning trees within the meta-graph based on that dependency tree. However, many different spanning trees exist for any given graph: we must further decide which of these we wish to find. Our decision is to select edges with the cheapest combined cost, thus producing a *minimum* spanning tree in the meta-graph.

Our reason for this is very similar to one of our reasons for producing a spanning tree: to prevent the repeated inspection of edges. Recall that any edge in a meta-graph is based on a path in the base dependency tree: any set of edges in the meta-graph thus indicates a unique set of paths within that dependency tree, with each path including a number of dependency edges. To truly achieve our goal of avoiding repeatedly observing relations between words, we must ensure that even in the dependency trees the collection of these edges contains as little duplication as possible.

While in a meta-graph all edge duplication is avoided by the use of a spanning tree, the situation is more complex when considering the dependency trees from which such meta-graphs are generated. The complexities are the result of the existence, in dependency trees, of unaligned nodes. Given these, paths may exist between different pairs of aligned nodes which nonetheless share an edge connecting two unaligned nodes, or connect one unaligned and one aligned node. This can be the case even if the paths between aligned node pairs would not result in cycles in the associated meta-graph, as illustrated in Figure 5.3.

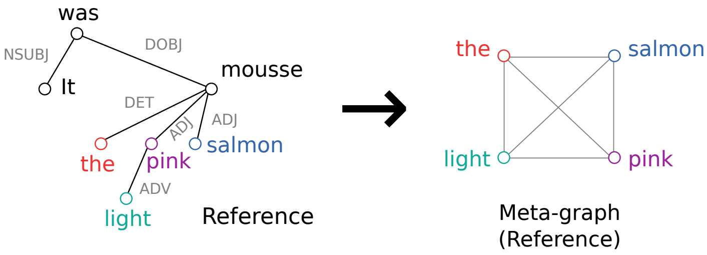


Figure 5.3: A sample sentence, using a similar visual convention to Figure 5.2, for which dependency edges must be considered multiple times for any spanning tree in the meta-graph

Reference: It was the light pink salmon mousse

In the example, three of the four aligned nodes – *the*, *pink* and *salmon* – are immediate children of a single unaligned ancestor, *mousse*. Notice that in the meta-graph, any spanning tree must necessarily involve at least one node being part of multiple edges, e.g. *(the,pink)* and *(pink,salmon)* both containing the node *pink*. In the dependency tree, the edge connecting such a node to the shared ancestor – *(pink,mousse)* here – must be considered in multiple paths.

Note that this example represents an edge case, requiring the parent of an aligned node to not itself be aligned, though this is far from rare. In cases when the parent of multiple aligned nodes is itself aligned, errors between that parent and its child(ren) can be considered separately from those between the parent and any nodes related to it via its own parent, as discussed in Section 5.4 and illustrated here by the nodes *light* and *pink*.

Specifically, in the case of any meta-graph edge such as *(light,salmon)*, whose dependency tree path includes all edge(s) from a meta-graph edge with a shorter dependency path such as *(light,pink)*, will almost exclusively have a cost equal to or greater than the shorter path. This is because any errors tied to dependency labels in the shorter path must necessarily also be included in the longer path due to its inclusion of the same edges: the only difference is that the longer path will contain more edges. While these edges may or may not increase the cost of the longer path, they cannot reduce it.

Given our goal of reducing as far as possible the repeated selection of individual dependency graph edges, we can take advantage of this relationship between short and long paths' weights. Simply put, a prioritisation of low-weight edges will avoid as much as possible any paths such as *(light,salmon)* which could also be represented by a series of shorter paths. In this manner, a minimum spanning tree in the meta-graph will usually represent the fewest possible dependency edges, while still ensuring that all aligned nodes are connected.

Note that exceptions exist to the rule that longer paths must cost at least as much as shorter sub-paths. Consider the example shown in Figure 5.4. With no mismatches between labels for any pairs of edges, only direction symbols may influence weights. The fact that *a* is before *Holmes* in both trees means that the direction symbols in both trees match for the longer path (*a,Holmes*), but the unusual swap of the position of *Sherlock* relative to *a* causes the shorter path (*a,Sherlock*) to require an edit operation to match direction symbols.

As direction symbols can necessitate a maximum of one edit operation, in order for a shorter path to be strictly more expensive than longer ones all labels must correspond for all relevant edges. This must also occur in situations where the order of two words in the sentence is exchanged. We consider the combination of these phenomena to be unlikely enough that such edge cases do not invalidate the powerful merits of cost minimality.

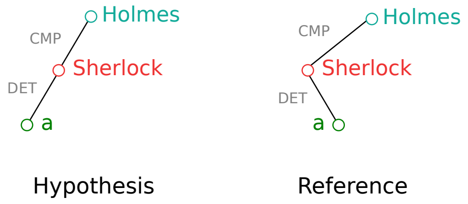


Figure 5.4: Sample dependency trees, using a similar visual convention to Figure 5.2, containing a two-edge path with lower L cost than a one-edge sub-path
Reference: a Sherlock Holmes
Hypothesis: Sherlock a Holmes

5.5 Singularity of spanning trees

We have thus far discussed our motivations for producing minimum spanning trees, and the mechanisms through which we generate meta-graphs and the weights of edges within them. Before we can approach the question of producing the spanning trees themselves, we must address the duality of our trees.

We remember that the edge weights in a meta-graph are defined in terms of pairs of nodes which are shared between two dependency trees. Both dependency trees are thus essential for any meta-graph, although we generate a meta-graph from the set of aligned nodes in just one of the two. As mentioned in Section 5.4.1, the dependency tree on which the meta-graph is based is selected through the ordinality of the alignment relation

C between the trees. Specifically, we stipulate that no node in one dependency tree D – the dependency tree whose node set N_D is used to generate the meta-graph – can be aligned to more than one in the other, D'.

However, it would not be meaningless to create two separate meta-graphs, one from D and one from D'. Indeed, given our goals of considering every node in both sentences, it would appear at first glance to be necessary to produce two such graphs in which to generate two separate spanning trees.

In this section, we demonstrate why this assumption is incorrect and only one meta-graph is needed. This greatly simplifies the problem we need to solve, whose discussion is continued in Section 5.6. To justify the simplification, for this section only we assume that two meta-graphs have been produced, based on the nodes from D and D' respectively and named M and M', and a spanning tree is produced in each. We show that the result is equivalent, in all respects which we consider important, to that of a single spanning tree in the meta-graph based on D.

We rely here on the assumption of strict one-to-many alignment ordinality between D and D'. This is done to simplify the explanation, yet in Section 5.7 we show that this assumption is in reality unimportant and the choice of base dependency tree can be arbitrary.

5.5.1 Mapping edge sets

We begin by observing a number of relative characteristics of the two meta-graphs. We can consider two edges, one in each meta-graph, to be related if the nodes connected by each are themselves aligned. More formally: for any pair of edges (n_M, m_M) and $(n_{M'}, m_{M'})$ in meta-graphs M and M' respectively, we consider that the edges are *counterparts* if $C(n_M, n_{M'})$, $C(m_M, m_{M'})$.

The assumption we have made about the ordinality of aligned nodes provides further information about counterpart edges. Recall our stipulation that no node in one meta-graph M be aligned to more than one in the other, M', coupled with the requirement by the definition of meta-graphs that all nodes must be aligned to at least one in the opposing meta-graph. Given an edge (n_M, m_M) in M, we thus have two possibilities for edges in M': there can be either zero edges $(n_{M'}, m_{M'})$ in M' if both n_M and m_M are aligned to the same node in M', or one if they are aligned to separate nodes.

Given a set of edges T_M in M, we can use this knowledge to generate a set $T_{M'}$ in M' of maximum size equal to that of T_M. We consider that due to the shared nature of the generation of their weights, the generation of $T_{M'}$ is a direct byproduct of the generation of T_M rather than a separate process.

Further, we consider that it would be unreasonable to produce two edge sets which were not related through this mechanism: it would be meaningless to include any edges in either T_M or $T_{M'}$ without similarly including edges in the other such that the two trees are related through counterpart edges. For example, to include in T_M the edge (*becoming,the*) in the reference meta-graph in Figure 5.2, one must also include in $T_{M'}$ the edge (*becomes,the*) in the hypothesis meta-graph as each edge is only meaningful in the context of the other.

5.5.2 Mapping minimum spanning trees

We now consider a special case of T_M, such that it fulfills the criteria stated above: representing a minimum spanning tree in M. We investigate which of these properties transfer to $T_{M'}$.

First, we trivially observe that the total weight of the edges in $T_{M'}$ must equal that of T_M. This is because the edge weights are defined through the alignments relation C, the same mechanism used to generate the set $T_{M'}$ itself. Thus, all one-to-one edge mappings must have the same weights. In the case of edges (n, m) in T_M which have no counterpart in $T_{M'}$, the definition of the weight $|(n, m)|$ forces the cost of such edges to be zero.

From this, we can show that the combined cost of T_M and $T_{M'}$ is minimal. Given that the weights of the two trees are equal, the total cost cannot be reduced in one tree without reducing it in the other. However, by definition the spanning tree T_M is minimal, so its cost cannot be reduced. If there can be no reduction in cost of either tree, their combined weight must be minimal.

Finally, we demonstrate that $T_{M'}$ must represent a connected graph in M'. Consider a situation which invalidates this: two nodes $n_{M'}, m_{M'} \in N_{M'}$ are not connected by any contiguous path in $T_{M'}$. We observe that their aligned counterparts $n_M, m_M \in M$ such that $C(n_M, n_{M'}), C(m_M, m_{M'})$ must be connected in T_M due to its nature as a spanning tree, with P_M representing the set of edges in the (unique) path between them.

For each edge (n_M, m_M) in P_M there are two possibilities, either of which allow us to consider that their counterpart(s) in M' are connected: either those counterparts are the same node, or they are connected by an edge which must be in $T_{M'}$ in order to allow (n_M, m_M) to itself be in T_M. Note that the contiguity of edges in P_M requires that all nodes be part of exactly two edges; as these nodes transfer uniquely to the counterpart nodes in M', the path in P'_M must similarly be contiguous. Thus, the set of all counterparts of edges in P_M must connect $n_{M'}$ and $m_{M'}$, and we have a contradiction.

With the properties of minimality and connectedness being shared between T_M and $T_{M'}$, we can consider that the two represent the best possible solution to our problem. Most importantly, the fact that both are connected graphs ensures that we have considered every error relative to every other, ensuring that no potential errors are ignored. The minimality of their combined cost, and the nature of T_M as a spanning tree without cycles, ensure that we have duplicated errors as little as reasonably possible.

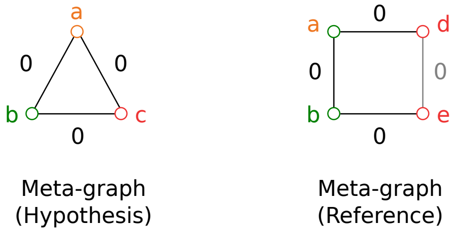


Figure 5.5: Sample meta-graphs, using a similar visual convention to Figure 5.1, with only black edges included in possible minimum spanning trees. Weights are indicated next to their respective edges.

Note that it is entirely possible for $T_{M'}$ to contain cycles, for the simple reason that it can contain fewer nodes than T_M. If multiple nodes in M map to single nodes in M', as we have allowed, nodes which are not directly connected in M may be directly connected in M'. As such nodes must be indirectly connected in M and thus also in M', this results in a cycle as illustrated in the synthetic example in Figure 5.5.

This situation represents a simple choice, as the possibility of duplication in M' is necessary to guarantee that all nodes in M are connected. We consider that ensuring that all errors in M have been considered, and thus allowing duplication, is more important than preventing duplication in M' at the cost of entirely ignoring node comparisons.

5.6 Producing spanning trees

Having demonstrated that by producing a minimum spanning tree in the larger of our two graphs M we have fulfilled our goals of representing every error at a minimal cost, we must now solve that simpler problem. We are able to utilise an existing algorithm to generate a set of edges forming a minimum spanning tree in M.

The algorithm we have chosen is the widely used Kruskal's algorithm, which has been shown to always result in a minimum spanning tree when applied to fully connected graphs with arbitrary positive weights [Kruskal, 1956]. It runs in $O(n\ log(n))$ time where n is the number of nodes in M [Nešetřil *et al.*, 2001], an important factor when calculating scores for large numbers of sentences.

5.6.1 Procedure

The (minimally) adapted version of Kruskal's algorithm which we use in DERP is shown in Algorithm 2, and described below.

Algorithm 2 Kruskal's algorithm as implemented for DERP

```
procedure KRUSKAL($N_M$)
    $edges \leftarrow \{\}$                              ▷ $edges$ is stored as a priority queue
    for all $n \in N_M$ do
        $set(n) \leftarrow$ unique value                        ▷ Initialise sets
        for all $m \in N_M : m \neq n$ do                  ▷ Initialise edge costs
            $edges \leftarrow edges \cup \{(n, m, |(m, n)|)\}$
    $cost \leftarrow 0$                                         ▷ Initialise cost
    while more than one set exists in $N_M$ do          ▷ Terminate when connected
        $\{n, m, weight\} \leftarrow$ pop cheapest element in $edges$
        if $set(m) \neq set(n)$ then                ▷ All moves affect disparate sets
            for all $o \in N_M : set(o) = set(m)$ do
                $set(o) \leftarrow set(n)$                         ▷ Unite the two sets
            $cost \leftarrow cost + weight$
    return $cost$
```

We begin by putting every node from N_M in a singleton set by itself and initialising the global cost to 0. We also create a set $edges$ of triples, with each element indicating two nodes and the cost of the edge between them.

We arbitrarily extract a triple from $edges$ such that the edge cost is the lowest remaining in $edges$ and its two nodes are not in the same set. We add the edge cost to the global cost and then unify the sets of both nodes, representing the inclusion of the edge in the appropriate spanning tree. We repeat this process until all nodes in N_M are in the same set. We then return the total cost of all selected edges.

5.6.2 Example

The weights of the edges in the meta-graph shown in Figure 5.2 are shown in Table 5.1, which also contains one possible order in which the algorithm could select edges.

We select the reference sentence to be the one from which we generate meta-graph M, as a result of its having multiple nodes aligned to a single hypothesis node. Three edges in the meta-graph have weight 0. Two of these, $($*Sherlock,a*$)$ and $($*detective,the*$)$, are simply paired with identical paths in the hypothesis tree, while one – $($*becoming,is*$)$ – has a default cost of 0 due to its two nodes being aligned to a single one in the hypothesis. These are the first three edges selected by the algorithm, their order being unimportant.

After the 0-cost edges have been selected, the cheapest remaining edges cost 2. These are $($*becoming,detective*$)$ and $($*becoming,the*$)$, of which the algorithm first selects the former. The latter is thus ineligible to be selected: nodes *becoming* and *the* belong already to the same set, representing the indirect path of already-selected edges (via *detective*) between the two.

The algorithm must then arbitrarily choose between $($*becoming,Sherlock*$)$, $($*becoming,a*$)$ and $($*is,a*$)$, as they share edge weight 3 while $($*is,detective*$)$ is ineligible through its nodes being already indirectly connected via *becoming*. Selecting $($*becoming,Sherlock*$)$ unites all sets and results in the spanning tree shown in Figure 5.6, with a total cost of $0 + 0 + 0 + 2 + 3 = 5$.

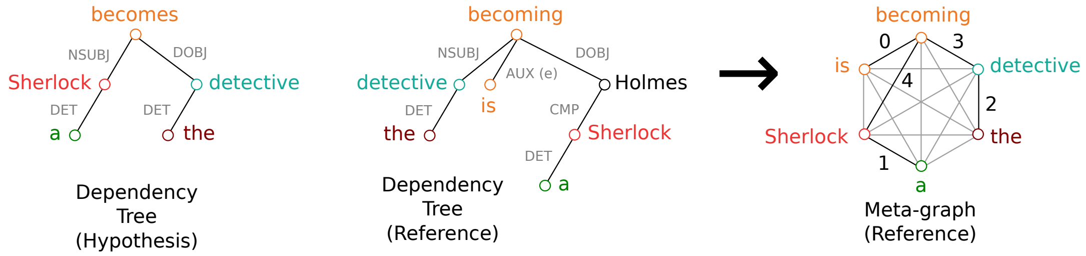


Figure 5.6: Dependency trees from Figure 5.1 with the meta-graph generated from the reference tree, using a similar visual convention to Figure 5.1. In addition to the information presented in Figure 5.2, the darker meta-graph edges indicate that they may be selected by the algorithm in the order described by their labels, as per the '#' column of Table 5.1. See the same table for the costs associated with each edge.

5.7 Other alignment ordinalities

Thus far, we have relied on a central assumption stated earlier: that no node in one of the two dependency trees may be aligned in C to more than one in the other tree. This

n_D	m_D	(n_D, m_D)	$(n_{D'}, m_{D'})$	L	L_{max}	#
becoming	is	$(,\beta,\text{AUX})$	*N/A*			0
becoming	Sherlock	$(,\alpha,\text{DOBJ CMP})$	$(,\beta,\text{NSUBJ})$	3	3	4
becoming	detective	$(,\beta,\text{NSUBJ})$	$(,\alpha,\text{DOBJ})$	2	2	3
becoming	a	$(,\alpha,\text{DOBJ CMP DET})$	$(,\beta,\text{NSUBJ DET})$	3	4	-
becoming	the	$(,\beta,\text{NSUBJ DET})$	$(,\alpha,\text{DOBJ DET})$	2	3	-
is	Sherlock	$(\text{AUX},\alpha,\text{DOBJ CMP})$	$(,\beta,\text{NSUBJ})$	4	4	-
is	detective	$(\text{AUX},\beta,\text{NSUBJ})$	$(,\alpha,\text{DOBJ})$	3	3	-
is	a	$(\text{AUX},\alpha,\text{DOBJ CMP DET})$	$(,\beta,\text{NSUBJ DET})$	3	3	-
is	the	$(\text{AUX},\beta,\text{NSUBJ DET})$	$(,\alpha,\text{DOBJ DET})$	4	5	-
Sherlock	detective	$(\text{CMP DOBJ},\beta,\text{NSUBJ})$	$(\text{NSUBJ},\alpha,\text{DOBJ})$	4	4	-
Sherlock	a	$(,\beta,\text{DET})$	$(,\beta,\text{DET})$	0	2	1
Sherlock	the	$(\text{CMP DOBJ},\beta,\text{NSUBJ DET})$	$(\text{NSUBJ},\alpha,\text{DOBJ DET})$	4	5	-
detective	a	$(\text{NSUBJ},\alpha,\text{DOBJ CMP DET})$	$(\text{DOBJ},\beta,\text{NSUBJ DET})$	4	5	-
detective	the	$(,\beta,\text{DET})$	$(,\beta,\text{DET})$	0	2	2
a	the	$(\text{DET CMP DOBJ}, \beta,\text{NSUBJ DET})$	$(\text{DET NSUBJ}, \alpha,\text{DOBJ DET})$	4	6	-

Table 5.1: Edge weights for the meta-graph in Figure 5.2. n_D, m_D are reference nodes, implying unique $n_{D'}, m_{D'}$ in the hypothesis. L and L_{max} are applied to $(n_D, m_D, n_{D'}, m_{D'})$. '#' indicates one possible order in which these edges' nodes' sets are united by DERP.

restriction, common among real-world alignment tools as discussed in Section 3.4.5, dramatically reduces the domain of our algorithm: it permits us to simplify the problem at hand to that of a spanning tree in a single meta-graph. While in theory it prevents a number of legitimate alignments, in practice such alignments are rare.

The most practical consideration behind our assumption is that it matches the alignment tools we use: these do not themselves generate many-to-many alignments. Indeed, the two systems we employ, GIZA++ [Och and Ney, 2003] and cdec [Dyer *et al.*, 2010], do not produce any more complex alignments than the strict one-to-many alignments assumed in most of this chapter.

To investigate the limitations imposed by this assumption, we first discuss the issues which would arise if we relaxed it, allowing many-to-one alignments in both directions or even many-to-many alignments. These relate to the requirement that each of the relevant sets of nodes be connected by DERP at a minimum overall cost.

To ensure that all nodes in both meta-graphs have been considered, we would need the DERP algorithm to consider the connected status of both simultaneously. While this is possible without modifications in the case of strictly (but bidirectional) many-to-one alignments, it becomes quickly intractable in the case of many-to-many alignments.

5.7.1 Bidirectional one-to-many alignments

In the case of one-to-many alignments which may occur in both directions, the algorithm still produces an optimal solution to the problem. To show this, we slightly rephrase our goal: we wish to determine the minimum total weight for a set of edges connecting all nodes in each tree.

We show this by imagining a separate post-processing step to the DERP algorithm detailed above. First, we arbitrarily select one of the two meta-graphs to be the M in which we generate a spanning tree. The resulting tree T_M must be minimal within M for the same reasons discussed earlier.

We observe that while T_M is guaranteed to connect M, the same is not true of its counterpart $T_{M'}$ in M'. If any node n_M in M is aligned to more than one in M', no edge in T_M involving n_M can connect more than one of those in M'. As a result, these last can never be connected together, preventing a spanning tree from being formed in M' due to the lack of full connectedness.

We can solve this problem by further adding edges in M' to $T_{M'}$, connecting disparate sub-trees within $T_{M'}$. Note that the proof of connectedness in Section 5.5 still holds in all cases except groups of nodes in M' which are connected to single nodes in M. Such nodes can be connected for free by adding edges in M' alone, through Equation 5.6.

Having augmented $T_{M'}$ with the additional edges required to connect M', we notice that we have now achieved our goal by connecting all nodes in both trees. In the process, we have not incurred any additional cost, as all extra edges we added had a cost of zero.

This means two things: first, the cost must still be minimal across both meta-graphs, as it has not increased from a value which was already a lower limit for a simpler problem. Second, the additional edges were not necessary in order to simply calculate the total weight as per our goal. The total cost calculated by the original DERP algorithm is thus equal to the minimum cost including the extra edges, meaning that the original algorithm is itself minimal in this case.

5.7.2 Many-to-many alignments

In the more complex case where groups of words in one sentence can be aligned to whole groups in the other, it becomes much more difficult to generate provably optimal edge weights. This is due to the multitude of ways in which groups of aligned nodes can be connected.

We first place a restriction on the weight generation we have discussed before, by requiring that in the case of many-to-many matches only, edges in only one tree cannot be assigned a weight of zero through Equation 5.6. This is because such zero-cost edges are a response to an edge case to which no other solution is intuitively reasonable: that of two nodes which have no distinct pair of counterpart nodes to connect to. With many-to-many alignments, this edge case is no longer applicable so its solution is not relevant.

If we nonetheless did not impose this restriction, the problem we face would become much simpler: a procedure similar to that in Section 5.7.1 could be applied to demonstrate continued optimality. This would in part be to connect all nodes involved in many-to-many alignments through zero-cost edges in their own trees. However, we consider this behaviour to be unrepresentative of the complexity involved in the sentences.

Having thus applied the above restriction, we consider the meta-graphs shown in Figure 5.7. Observe that any node in either meta-graph can be matched with either other node in its own, with that edge being partnered through edge pairs with any of the three edges in the opposing meta-graph. This leads to multiple possible weights for each edge, each calculated by comparing the path with that of a different pair of counterpart nodes.

This gives rise to an exponential number of possible edges for the algorithm to calculate: with three nodes in each tree in the example, this gives rise to $3^2 = 9$ possible edge pairs for which the algorithm must calculate path disparities. While this is unlikely in practice to make the algorithm intractable, as large phrases are likely to have smaller sub-phrases which can be aligned together, it nonetheless adds complexity.

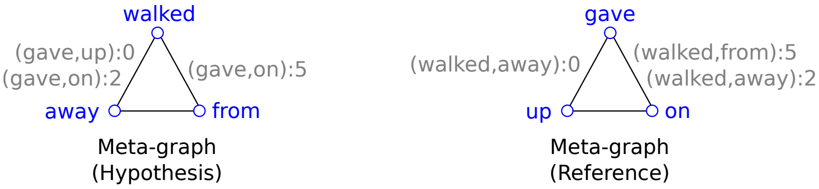


Figure 5.7: Example meta-graphs with all nodes in each tree aligned to all in the other. Missing weights cost 6 edit operations; all cheaper weights are shown.
Hypothesis: walked away from
Reference: gave up on

A much more severe problem than simple computational complexity is that DERP is not guaranteed to produce a solution which is either minimal or connected in both trees. We first demonstrate the latter claim.

Selecting either meta-graph as M, it may be possible to produce a spanning tree in that meta-graph while adding only one edge multiple times to $T_{M'}$ in the opposing meta-graph. For example, to connect the Reference meta-graph in Figure 5.7, the algorithm would select the edge pairs *(walked,away,gave,up)* then *(walked,away,gave,on)*. While the algorithm would terminate after this, the node *from* would be left unconnected in the Hypothesis.

This limitation could be mitigated by simply continuing to add edge pairs to either T_M or $T_{M'}$ until both sets represent spanning trees. However, such behaviour would prevent the solution from being provably minimal, as we can show in the same example. After selecting the above two edge pairs, the algorithm would continue to select *(walked,from,gave,on)*, resulting in a final cost of $0 + 2 + 5 = 7$. However, a cheaper solution would be to select only *(walked,away,gave,up)* and *(walked,from,gave,on)* for cost $0 + 5 = 5$.

To detect the optimal solution in this case, the algorithm would need to be dramatically extended. This would likely involve incorporating nondeterminism: remembering different alternatives for selecting edges and finally selecting one only once all have been considered. In any case, it would significantly increase the computational complexity of the algorithm, making it likely to be unwieldy in practical scenarios.

5.8 Aggregation & normalisation

Having produced a spanning tree in one of our meta-graphs, we can consider the integer cost of that tree to be a score for the given sentence pair. However, such a cost is not as versatile as we would consider necessary for an evaluation metric for machine translation, for one important reason: it cannot be easily compared between sentences. This is a result of the cost being directly related to the number of edges in our spanning trees, which in turn is a function of the number of words in the sentence. Thus, the ranges of scores possible for sentences with different lengths can vary dramatically, with ‘high’ costs for a short sentence being potentially trivially low for a longer one.

We address this problem through a process of normalisation, which involves dividing the total cost produced by the algorithm by a sentence-specific *normalisation quotient*. There are a number of possible methods of producing these quotients, each producing subtly different final scores.

5.8.1 Word count

The most obvious method of producing a quotient which controls for the length of the sentence is simply to use either the number of words itself or the number of edges in the spanning tree. These quotients would result in a score representing the average Levenshtein distance across all edges. This approach is intuitively meaningful, but has two issues.

The more minor of these is that such scores are not in the same [0,1] range as the vast majority of existing scores in our field. Working with unbounded positive numbers in some cases but with real numbers strictly in the range [0,1] in others makes comparison between tools a little more confusing. While this is mathematically unimportant, we consider consistency with our peers to be an important consideration.

The second and more DERP-specific limitation of using sentence length as a normalisation quotient is that it still includes an element of structure in the final score, confusing comparison between sentences.

Consider two sentences: S, containing sequences of unaligned nodes in the paths between each pair of aligned nodes, and S', with the same numbers of aligned and unaligned nodes but for which all unaligned nodes are located in separate subtrees from aligned ones. Even if similar numbers of dependency edges fail to match between the two trees, the meta-graph based on S' will likely contain many edges with low but non-zero costs while S will result in a smaller number of meta-graph edges with higher costs. Dividing by the number of aligned nodes would result in S appearing to have a significantly higher error than S', while dividing by the total number of aligned and unaligned nodes would instead skew the result in favour of S.

This disparity can be seen in the examples in Figure 5.8. Dependency labels for edges are omitted: we imagine, purely for simplicity, that each tree is compared to a hypothesis tree with identical structure but entirely mis-matching labels, resulting in total edge weights of 6 and 3 respectively for the two sentences.

In the first, using a normalisation quotient equal to the number of edges in any spanning tree in the associated meta-graph (always one less than the number of nodes in the meta-graph, thus 3) would result in a final score of $6/3 = 2$. While this may seem appropriate, applying a similar process to the second sentence would produce a score of $3/3 = 1$, a much lower value. Similarly, using the number of edges in the dependency trees rather than the meta-graphs would result in scores of $3/6 = 0.5$ and $6/6 = 1$ respectively.

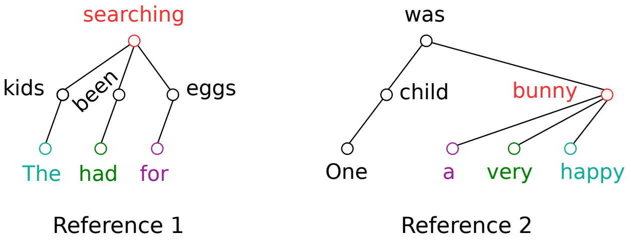


Figure 5.8: Sample dependency trees, using a similar visual convention to Figure 5.1 but with dependency labels omitted
Reference 1: The kids had been searching for eggs
Reference 2: One child was a very happy bunny

5.8.2 Worst-case paths

For the reasons outlined above, we reject the simple use of sentence length as a normalisation quotient. Instead, we rely on the maximum errors in each individual aligned node pair, through L_{max} (Equation 5.4). These control for all features which contribute to the Levenshtein distance, including both sentence length and the lengths of individual paths.

Our method of using L_{max} to produce a normalisation quotient is linked closely with our method of generating spanning trees. Each time we calculate weights for an edge in a meta-graph, we apply L (Equation 5.3) to calculate the Levenshtein distance between the features of the associated paths in two dependency trees. For each such calculation, weights can have any value between 0 and an L_{max} based on the paths. This theoretically allows us to normalise each edge weight separately, by dividing the real difference L by its maximum.

In practice, normalising edge weights before the production of spanning trees would undermine many of the assumptions we rely on. Notably, we have discussed how the selection of a minimal tree prioritises short paths over longer ones with potentially more errors. Longer paths, however, could receive lower normalised scores than shorter alternatives containing fewer errors but also fewer error-free edges.

For example, an L result of 1 for the dependency edge *(a,becomes)* in Figure 5.6 would make it equivalent to the shorter *(Sherlock,becomes)* without normalisation. However, the former's length would cause L_{max} to be 3 relative to the shorter path's 2: this would result in a normalised score of $1/3$ relative to $1/2$, causing the algorithm to strictly prioritise it. This is the opposite of our intended behaviour, which would be to consider the lack of error in *(a,Sherlock)* only once: when selecting the edge pair *(a,Sherlock,a,Sherlock)* for cost 0.

Instead of applying normalisation before generating spanning trees, we thus consider it at the end of the algorithm. Each time an edge pair is selected by DERP, its weight (L) is added to the global final cost for the sentence. In addition, we record the maximum possible weight (L_{max}). Once a spanning tree has been completely generated, we then divide one by the other to produce a final score. This will necessarily produce a value in the desired range [0,1].

We also subtract this value from 1 to ensure that a higher value is better, to match our intuition. Given a complete spanning tree with weights generated from edge pairs in the set T, we thus calculate a final score for DERP as follows:

$$score = 1 - \frac{\sum\limits_{e \in T} L(e)}{\sum\limits_{e \in T} L_{max}(e)} \tag{5.7}$$

5.9 Implementation

Having designed an algorithm which is intended to represent a reasonable and granular solution to the question of evaluating the word order of machine translation sentences through investigation of structure, we have implemented that algorithm to allow for empirical investigation. Similarly to DTED (Chapter 4), we have produced a prototype system using the Python language, built using the NLTK libraries [Bird, 2006].

As the generation of dependency trees requires both a trained dependency parser and a tagger system which may be separate, we have used the same configurations as for our earlier tool. Described in more detail in Section 3.4, these are: running the Stanford Parser [Klein and Manning, 2003b] with its own internal tagger, and the Malt Parser [Nivre, 2003] with both the tagger included in the Stanford Parser and the maximum-entropy part-of-speech tagger bundled with NLTK [Malecha and Smith, 2010].

In addition to the tags and dependency parses produced by these tools, the alignment relation C is a key input to DERP. We have generated this using two more external tools: the widely-used off-the-shelf tool GIZA++ [Och and Ney, 2003], and the word alignment component of the cdec project [Dyer *et al.*, 2010] from Carnegie Mellon University.

To calculate Levenshtein distances between paths, we have used Python's 'Levenshtein' package v0.12.0 [Necas and Haapala, 2014]. As that package uses characters as atomic units rather than words, we first convert dependency labels into unique but otherwise arbitrary characters. The same character is used for every occurrence of any given dependency label, while two dependency labels are guaranteed to be represented by different characters.

5.10 Flattened version: 'f' flag

In addition to running the main DERP algorithm described above using each combination of the above tools, we have run an altered, simplified version on *flattened* trees. As described in Sections 3.4.4 and 4.4, our goal with this version of the tool is to investigate its accuracy and reliability when deprived of the information provided by the dependency tree structure.

Flattened DERP is run in a manner similar to flattened versions of DTED: by providing a simple 'f' flag when invoking the algorithm, we instruct it to perform a preprocessing step to remove the structure provided by the dependency parsing. This discards all information offered by the parse which is not present in the original unparsed sentences: each node is set to be the (only) child of its immediate predecessor, with all dependency labels replaced with the uniform (and thus meaningless) label DEP.

5.10.1 Features

While flattened DERP contains none of the structural information for which the algorithm was designed, it nonetheless retains significant interesting features. Specifically, the paths we inspect continue to indicate two key descriptors of the relationships between words: distance and direction.

The distance encoded in a flattened path is extremely straightforward to understand: any path between two nodes directly indicates the number of words between them in the sentence; any mismatch thus indicates that the relative positions of a given pair of aligned words are not identical, while the resulting weight also provides a simple measure of the disparity.

In addition, directionality is encoded in two important ways. First, the direction symbols α and β are unaffected by the flattening process, in that a mismatch between two paths, with one linking a node and its ancestor and the other linking a counterpart node and its descendant, will be encoded as rightward and leftward directions respectively and thus immediately incur one edit operation to match. In addition, the sequences of dependency labels in each path would be encoded in separate elements of the 3-tuples described in Section 5.3.1, resulting in maximal edit operations for such edge pairs.

5.10.2 Comparison with Kendall’s τ

As discussed in Section 4.4.1, at least one flattened version of DTED can be compared to the Kendall’s τ algorithm for counting relative mismatches; such a comparison is even more striking in the case of DERP. Kendall’s τ, described in more detail in Section 3.5.2, inspects the number of mismatches between nodes’ relative orders within a pair of sequences (e.g. two sentences). It does this by comparing every node to every other and considering them correctly or incorrectly positioned depending on whether the nodes are in the same relative order in both sequences.

Kendall’s τ thus performs a similar task to the direction symbols in flattened DERP: it observes mismatches in relative order and calculates a penalty based on those mismatches. The algorithms significantly differ in one major respect: their method of considering the severity of any individual error without allowing one (in)correct node to disproportionately skew the overall score.

Kendall’s τ accomplishes this by placing the errors it detects in the context of the number of pairs of nodes whose relative positions match between the two input sequences. The algorithm’s final output thus indicates the proportion of node pairs whose orders match, relative to those which do not. In this way, individual nodes’ errors may be observed multiple times – when comparing the node to several others – while the potential for a single node’s error to propagate throughout the score for an entire sentence is mitigated.

DERP uses a different technique for placing the errors it detects in context, limiting the impact of any individual node by ensuring that it is compared with as few others as possible. This is the goal of the spanning tree in the relevant meta-graph. In this manner, DERP assesses the severity of any given node’s error in a single operation, by comparing paths to result in a limited number of edges which contribute to the final score. This limits the impact of any given word on the overall sentence score, without omitting any information as the type and severity of the mismatches are encoded in the weights of such edges.

5.10.3 Expectations

Given the above comparisons, we expect flattened DERP to perform at a similar level to Kendall's τ when compared with human judgments, as the two algorithms perform similar tasks. Both are expected to predict human judgments better than the more naïve DTED. A somewhat simplistic summary of the differences between flattened DTED, flattened DERP and Kendall's τ could be as follows:

DTED: While DTED is capable of observing which nodes are incorrect, it is limited to simply applying close-to-uniform penalties to these, without taking into account the distance by which words have been displaced.

τ: Kendall's τ begins to take distance into consideration by considering each node in the context of each other: severe errors with individual nodes will thus be observed multiple times.

DERP: In addition to observing individual errors and quantifying their severity, DERP attempts to isolate errors for individual words and consider them with as much context as possible.

While flattened DERP has direct parallels among existing literature in the form of Kendall's τ, the original, non-flattened version is somewhat more unique. As such, it is difficult to predict its performance relative to Kendall's τ and its own flattened version. The relative performance of these three systems will greatly inform our understanding of the relevance of structure in machine translation.

Chapter Six

WOJ-DB: a Word Order Judgement DataBase

6.1 Introduction & motivation

We have thus far described how to evaluate the word order of machine translation using automatic techniques. We have attempted to justify our design choices using theoretical analysis of our algorithms, and by drawing parallels between those algorithms and existing related ones in literature. The goal of all these algorithms is to evaluate the quality of the sentences on which they are run.

To determine the viability of the approaches we have used, we must evaluate the correctness of the scores generated by our metrics; the normal method of doing this is to compare them with a gold standard. In translation, the only gold standard we can accept is that of judgments given by humans.

Any database of quality judgments on translations can inherently provide two separate benefits. The first of these is the evaluation, through comparison against a ‘correct’ measurement, of automatic quality evaluation tools. The second is deepening our insight into human evaluation in practice, which can guide future research in both the areas of translation and of evaluation.

A number of quality rating databases exist already, providing a wide range of language pairs, quality ranges and judgment types. These are detailed below, but all systems we know of have one feature which makes them ill-adapted to our needs: they all provide overall judgments on the sentences, considering all factors together to summarise their holistic quality. Given that our own metrics are designed to measure specifically the quality of the word ordering, we need a more bespoke database. As such we have performed a survey, gathering judgments of both holistic and order-specific quality from local native English speakers.

6.1.1 Existing human judgments

Despite the shortcomings of existing evaluation criteria, we have nonetheless made use of sentences included in such evaluation corpora as the foundation of our survey, and we have incorporated the judgments provided in a limited manner into our evaluation process. We considered several such corpora before ultimately selecting WMT as our basis, as described below.

NIST judgments

The American National Institute of Standards and Technology, NIST, has compiled a set of corpora which are intended to be used for evaluating automatic metrics for machine translation. Available through the Linguistic Data Consortium [2011], we have obtained 73 sets of translations. These are divided into 31 sets translated from Arabic into English and 42 sets translated from Chinese into English. The corpora range from 183 to 267 sentences per set, with a total length of 16107 sentences.

These translations are taken from a general news domain, with translations ranging quite significantly in quality. The evaluations associated with these sentences are adequacy scores, with integer values ranging from 1 to 7.

NTCIR judgments

Provided by the Japanese group NTCIR, a number of corpora of translated sentences, along with adequacy scores for each, are available for public use with permission [Goto *et al.*, 2013]. While a number of different data repositories exist, each with different contents and goals, we have obtained the 'PATMT' corpora which were released in 2007, 2008 and 2009. These consist of a total of 40 sets of translations, each translated by a different system and containing 299 sentences for those from the latter year and 100 for those from either of the former.

These translations are from the technical domain of patents, with the quality of the translations again varying widely. Scores are provided in the range 1 to 5, with multiple judges providing scores on each sentence.

Terra

While the judgments mentioned thus far relate to individual sentences, some corpora exist which provide details about the problems with individual words. The Terra corpus [Fishel *et al.*, 2012a] is one of the earliest of these to relate specifically to machine translation errors rather than human second-language learners'. It consists of several hundred sentences annotated with the categories of errors associated with each word, if any.

The categories used in Terra are based on the classification system of Vilar *et al.* [2006b], including missing words, incorrect word forms and, most relevantly to us, improper word ordering. The level of detail included in the dataset is highly interesting for a number of different evaluation scenarios, but is not quite appropriate for our own.

As our tools are intended to measure the sentence-level overall impact of all incorrectly ordered words, it is difficult to determine a truly comparable 'correct' score for this based on information in Terra. While the quality of word ordering in a sentence may have some direct relationship to the number of incorrect words, contextual factors governing the importance of such errors make such a relationship hard to predict.

As such, we do not feel we can confidently extrapolate a general sentence-level score based on individual annotated words. This leaves Terra's judgments not directly comparable with those of DTED or DERP, preventing it from being an appropriate dataset with which to judge our tools.

The TaraXÜ corpus

As a result of the machine translation community's growing interest in more granular, error-specific analyses of machine translation (see Section 2.3), the TaraXÜ project aims to involve human experts to a greater extent in the development of machine translation in a broadly similar manner to Terra. Presented by Avramidis *et al.* [2014], it builds on a number of smaller projects [Popović *et al.*, 2013; Avramidis *et al.*, 2012] to provide sentences with a wide variety of annotated quality information.

It contains sentences ranked against each other like in many other datasets, but additionally provides descriptions of the exact errors present in the sentences. These can be broad descriptors of the most serious issues with a sentence, which can be from 7 categories including word ordering, or word-specific errors of 8 types which do not reference word ordering. The dataset also includes a limited number of sentences which have been post-edited by hand to match the meaning of a reference [Avramidis *et al.*, 2014].

While the data presented in the TaraXÜ corpus is impressive in its scope and detail, it does not address the exact question we are investigating: the quality of word ordering. The binary information it contains, indicating whether order is among the most important error in each of almost 1500 sentences, is not sufficient for our needs: similarly to Terra's word-level judgments, it is simply not directly comparable to the scores produced by DTED and DERP.

WMT judgments

Since 2006 [Koehn and Monz, 2006] the Workshop on Machine Translation (WMT) has provided evaluation mechanisms for machine translation. Specifically, scores have been calculated for machine translations produced by novel systems submitted to the Workshop as part of its Translation Task. These scores include automatic metrics, such as BLEU, but also subjective overall quality as rated by human judges. These human judges have for practical reasons primarily been the same people who submitted tools and translations to the Workshop [Koehn and Monz, 2006], augmented by volunteers and paid professionals.

The evaluation criteria have changed over the years, starting with simple adequacy and fluency ratings for individual sentences [Koehn and Monz, 2006]. Over time these have been replaced with general relative rankings indicating sentence quality [Bojar *et al.*, 2014, 2015, 2016a]. In each year, scores were provided across small groups of sentences through the use of the open-source system Appraise [Federmann, 2012].

Judges in this environment were given up to five output sentences at once and asked to rank their relative quality, allowing ties. These judgments were then processed as pairwise translation comparisons and evaluated using a modified version of the TrueSkill algorithm [Sakaguchi *et al.*, 2014]. More recently, a form of direct assessment has been introduced, with absolute judgements being assigned to sentences [Bojar *et al.*, 2016a], but this approach is in too provisional a phase for us to consider using its scores.

Given that the entire set of sentences submitted by all systems as part the annual Translation Tasks has been eligible for evaluation, the set of judgments includes a vast range of qualities, translation techniques, source and target languages. Note that while WMT has in the past offered a number of domain-specific translation corpora, we only use those which are taken from news sources, to ensure the sentences cover a broad range of topics and use accessible language.

The WMT dataset also has a significant benefit for our purposes in that the sentences provided as part of the corpus are not preprocessed in any way. While other corpora which we have obtained provide tokenised sentences which may look incorrect or confusing to lay readers, the WMT sentences are provided in the format in which they are found – allowing them to be given out equally appropriately to uninformed judges and machine evaluation tools.

However, they have the significant downside that they are not strictly absolute quality measures, but instead are relative ranks. This is for a number of strong reasons [Callison-Burch *et al.*, 2007], but causes the ratings to be difficult to compare objectively: for example, a sentence ranked '2' could be the second-best out of five sentences representing a range of qualities, or could simply be considered *not quite as bad* as three other sentences which the judge considered to all be marginally more unintelligible than it.

Despite the inherent downside of using ranks rather than absolute scores, we have considered that WMT judgments are the most suitable for our needs out of the available options. With the wide range of quality and techniques used by the systems submitted to the Workshops, and their abundance over the past several years, we consider the translations produced for them make an excellent basis for scientifically varied and thus worthwhile investigations. The number of sentences, source languages and translation approaches involved in WMT are unmatched in the other datasets we have described. The rank judgments, while not ideal for our purposes, provide a tentatively helpful comparator when evaluating any judgments we provide.

In order to provide a scoreset which is as homogeneous as possible, we have normalised WMT judgments to provide scores in the range [0,1] similarly to those generated by our and other metrics. For each rank s within a set of ranks S – that is, each sentence in a group of up to 5 sentences considered by a single judge at one time – this is done by applying the following calculation. Note that in recent years [Bojar *et al.*, 2016a] multiple hypotheses from different systems may be ranked simultaneously if their text is identical: such sentences are grouped together through our calculations, all receiving the same score as if there were only one $s \in S$.

$$score_s = 1 - \frac{rank_s - 1}{|S|} \tag{6.1}$$

6.1.2 Word order judgments

As mentioned above, while the above datasets are very useful for understanding general translator goodness, and even when investigating more detailed information such as the individual errors in a sentence, they do not provide the most relevant judgments to our project: an indication of how severe the word ordering errors in a sentence are.

This specific evaluation criterion cannot be guaranteed to be closely matched with holistic judgments, or word-level error types. To our knowledge, no corpus has to date been gathered which focuses on it precisely. We have thus produced WOJ-DB, a corpus of human judgments of the quality of the word order in the given sentences.

6.2 Overview of WOJ-DB

WOJ-DB is an attempt to provide a comprehensive database which may be used for any project investigating the significance of word order errors in sentences. We have taken a wide range of sentences from recent years of WMT, and have produced questionnaires from them which have been given primarily to native speakers of English. These questionnaires contain both genuine machine translations and automatically generated sentences designed to highlight certain aspects of translation errors. All sentences we use have originally been taken from recent years of WMT and thus demonstrate hugely varying qualities, translation techniques and source languages.

These questionnaires provide pairs of hypothesis (machine-produced) and reference (human-produced) translations, and ask judges to rate each hypothesis on two standard five-point Likert scales [Likert, 1932]. First, they are asked for a simple adequacy rating, describing the extent to which the correct meaning of the sentence is still clear. Second, they are asked to quantify how much the ordering of the words in each sentence affects that adequacy rating.

In each questionnaire, 50 sentences are included, split into four categories: machine-translated hypothesis sentences which exist in every questionnaire; machine-translated hypothesis sentences which are unique to the given questionnaire; and automatically generated permutations of reference translations which also exist in either one or all questionnaire(s).

The scores for all of these, along with extensive metadata about the sentences and the judges, have been bundled into a single dataset which can be used to evaluate automatic scores.

In designing such a database, our goal was to provide a resource which may be useful to future researchers in the broad area of evaluation of word ordering. With this in mind, we have attempted to make it as generic as possible, including broadly phrased questions and a wide variety of sentences. We have also provided various methods of evaluation of the scores themselves, to ensure any conclusions drawn from them are not dramatically affected by systematic differences between individual judges.

6.2.1 Flexibility

In order to provide as generic and widely applicable a corpus as we could, we have attempted to keep our dataset both simple and as extensive as possible.

We have ensured that WOJ-DB will be applicable to as wide a variety of other experiments as possible by providing extensive metadata on our sentences and our participants. While exact details can be found in Sections 6.3.2 and 6.6, broadly we have asked for extensive information on participants' language experience along with a number of general descriptors like age and gender. For sentences, we have stored information such as their length, source languages and percentage of words which could be aligned using our two alignment tools.

While the amount of metadata makes our database applicable for many different tasks, its simplicity is first and foremost represented in the questions we ask of our judges. We have chosen to ask judges to provide absolute quality scores, rather than compare between different sentences, to permit the sentences to be considered in isolation in both our own and future experiments.

While this has been shown to make scores more difficult or biased [Callison-Burch *et al.*, 2007], we have attempted to minimise the negative impact of the choice. As discussed by Bojar *et al.* [2016b], the presentation of both comparison sentences in English rather than in a source and target language, and the judgment of sentences in isolation (i.e. through random sentence selection) instead of in a broader context of a specific translation system, both contribute to the reduction in bias relative to ranking; both are the case in our survey.

For each pair of sentences with which participants are presented – first a machine-produced hypothesis translation then a human-produced reference – they are asked to provide scores, ranging from 1 to 5, for the following two questions:

1. How difficult would it be to grasp the meaning of the second sentence if you were only shown the first?
2. How much does the ordering of the words on its own cloud the meaning of the first sentence?

Each of these two questions has been designed to represent fundamental qualities of the sentence. The first asks about adequacy, the extent to which a given translation communicates the sense of the source sentence (see Section 3.3.2). We assume in all cases that the reference translations contain all pertinent information from the source, and can thus act as surrogates. While this assumption will not always be correct, we consider that it is a worthwhile simplification to make, as it allows judges with only one language – English – to provide meaningful judgments on sentences coming from a wide range of source languages.

The second question relates to the specific quality we are investigating: word ordering. This question is intended to be as explicit and simple as possible, with deliberately no reference to structure or specific aspects of the ordering. We have allowed participants to decide for themselves how much any given type of error affects the sentence as a whole.

Our decision to include an adequacy rating at all, rather than to ask only about the ordering of the words, was considered necessary for multiple reasons. First, it allows us to observe the performance of our tools when predicting overall quality as well as word-order quality, in response to the research question put forward in Section 3.2.3.

The second reason for such a question is that its existence allows those scores to be used for a much wider set of projects than if it were not included. For example, having ratings for general features of the sentences as well as specific ones allows us to investigate the relationship between the two through our research question of Section 3.2.3: by inspecting the similarities between the two sets of judgments, we can glean information about the impact of varying qualities – such as word order – on the broader comprehensibility of the sentence.

Both questions are intended to be simply worded, directly understandable to any native English speaker. This was to allow a broad range of participants, most with little or no linguistic experience, to contribute their opinions. The avoidance of professionally trained linguistic workers is intended to make the judgments closer to natural and uninfluenced intuition, and thus also closer to the broader opinion of the general public.

Continuing this goal, the participant pool we recruited from was as broad as feasible given our circumstances. Situated within the small University-dominated town of St Andrews, the vast majority of those living nearby were well-educated, while many – especially those interested in participating in academic surveys – were British students aged approximately 18-25. However, by advertising to the entire University body and using word of mouth, we attempted to mitigate this limitation and recruit judges from as wide a set of demographics as possible. For a more detailed breakdown of participant backgrounds and recruitment techniques, see Section 6.5.

Other than through their exact wording, we have attempted to make our questions more broadly applicable by varying the types of sentences used in the survey. We have ensured a range of qualities and error types in two separate ways: first by taking our sentences at random from several years of data from WMT, and second by including artificial permutations of human-produced reference translations. More detail on the former can be found in Section 6.1.1, while the latter is described in Section 6.2.3.

Our choice of requesting responses using Likert scales was another way to reinforce the wide applicability of our survey. We rejected more categorical answer formats, such as providing a descriptive word for each of the separate quality measurements, as such specific labels could have unduly influenced participants' choices in ways which could not be measured: for example, depending on the participant words like 'confusing' and 'unclear', two words which were considered as labels for two points on such an explicit scale, could have been interpreted as either very similar judgments or, equally, far apart – depending only on the exact linguistic perspective of the untrained judges. It would thus be very difficult to map such words onto an unlabelled absolute scale such as those of our automatic metrics.

We also rejected more continuous scales, such as a visual line which could be marked by the user, for similar reasons: given such a wide choice range, it would be difficult to interpret the meaning of a specific pair of judgments: for example, would the participant consider two judgments which differed by 10% of the length of the line to be dramatically different, or should those judgements even be interpreted as equal? The answer to this question would likely change for each participant, and could even be highly unclear from the participant's own point of view.

Another possible scoring mechanism would have been to ask participants to rank hypothesis translations relative to each other, in a manner similar to WMT. However, while this could have made our data more easily comparable to WMT data, and thus more verifiable, it was considered not worth the significant sacrifices it would have entailed. First, confronting untrained (or even trained) readers with a series of only marginally different or difficult-to-compare sentences may have been too challenging a request, leading to wildly varying scores between and even within participants [Denkowski and Lavie, 2010]. Second, relative ranks would have been much more difficult to interpret in the context of the absolute scores provided by our tools DTED and DERP, and other comparable tools such as BLEU and Meteor.

6.2.2 Verifiability

Our second design priority for WOJ-DB was to ensure that the data it contains are as reliable as possible. Success in this strengthens any conclusions we, or future researchers, may draw from it. We have thus built in a number of checks and balances into our data collection, by which the scores themselves can be evaluated and adapted. The first and foremost of these is that we have included extensive metadata on both participants and sentences; as stated earlier, full details of this metadata can be found in Sections 6.3.2 and 6.6.

Verifiability was another goal relevant to the choice of including a broader 'adequacy'-focused question within our survey. Our fundamental aim in producing WOJ-DB is to create a trustworthy and usable database of word order judgments, however simply providing such judgments on their own would provide little link between WOJ-DB and more commonly available holistic score types. By asking judges to provide two assessments of each sample, we provide this link as we – or others – may assess the appropriateness of any individual judge's scores by comparing those with other trusted scores. Unfortunately, the negative side of our choice to use data from WMT is that the available trusted scores are not directly comparable, being ranks rather than absolute judgments on a specific feature of the sentences. Despite this, we feel we can expect a certain level of similarity between the two.

A second method of verifying the scores we obtained was to vary whether or not sentences were shared between different participants' questionnaires. While we desired to make WOJ-DB as extensive a dataset as possible, providing as many individual judgments as we could, we considered that it would be close to meaningless to simply ask each judge to provide assessments of unique sentences, with no corroboration between participants. Given the subjective nature of translation quality, our judges were expected to necessarily vary on their ways of interpreting meaning and of assessing the severity of errors. Such differences are mathematically well-understood, having the technical name of 'random effects' within statistical literature [Moulton, 1986].

Assuming the presence of such random effects, we need to be able to observe the differences between how various judges scored certain sentences. Our means of doing this was to include the same 20 sentences in every single survey, thus making up 40% of each questionnaire. We felt that 20 sentences per user were an adequate base from which to mainly calculate the relevant effects, while leaving 30 unique scores per scorer to provide the bulk of the data. We also took the opportunity to manually select which sentences would be included in every survey in order that every participant was provided with a range of types and severities of errors within the sentences they judged. More detail on the shared sentences can be found in Section 6.3, while calculated random errors are described in Section 7.2.3.

Our final means of verifying the relevance of our data was to include sentences which, rather than being meaning-focused translations, are instead intended to simply represent specific error types with varying degrees of severity. We have called these sentences 'automatic permutations', as they involve randomly permuting human-produced reference translations according to a number of simple algorithms.

6.2.3 Automatic permutations

The goals of such permuted sentences are twofold. First, by producing sentences which have errors whose type and severity can broadly be known in advance, we can observe and statistically evaluate the extent to which judges do indeed detect such errors: we thus provide a quality control for our human evaluators. Specifically, automatic permutations are associated with a 'degree' descriptor. This is simply an integer indicating how many words in the sentence have been affected, representing an approximation of the severity of the errors it contains. Just like the WMT ranks for 'real' translations, this can then be statistically compared with each judge's scores.

It is once we decide that our judges are reliable, however, that automatic permutations become most interesting. The algorithms by which they are generated are deliberately simplistic, in order to allow them to be easily understood as atomic components of more complex errors – errors which, in real translations, may be too complex to generalise about. While a normal translation may produce an error which is classified by our judges as relating to word order, beyond that classification a low 'word ordering' score tells us little about exactly what went wrong with the sentence and why it was significant.

However, in the case of an automatically permuted sentence, we will in such a case know exactly what type and quantity of basic changes were made, and can thus begin to evaluate the relevance of such changes to the judge's overall impression.

We have for these purposes generated four types of automatic permutations, named 'order', 'swap', 'phrase' and 'choice' respectively. The permutations were generated in all cases by taking human-produced reference translations from the existing WMT corpus, and applying simple alterations to produce new sentences; these alterations are described in the following section. The resulting sentences were then included in our survey alongside the machine translations, with scores gathered and analysed using identical techniques.

Generation procedure

More specifically, the generation process was as follows. First, a temporary corpus was created from sentences from all the years of WMT from which hypothesis sentences were taken: 2014, 2015 and 2016. From each corpus, exactly 450 reference/hypothesis sentence pairs were chosen at random: this was to ensure all corpora contributed equally to the permutations, as individual contributions to WMT for the chosen years ranged from 497 to 3000 sentences in length. More details on the corpora used can be found in Table 6.2.

For each of the sentences chosen, each of the four permutation algorithms were applied multiple times. Each algorithm includes the current 'degree' as a necessary parameter, indicating how many words are to be manipulated in total. Certain degrees will necessarily not make sense for certain sentences: for example, there may not be enough words in a sentence to permute up to a high degree, while 'swap' permutations are applied to pairs of words so necessarily cannot exist for odd-numbered degrees.

All degrees from 1 to 18 were tried for each of the four algorithms on every sentence; after manual inspection of initial results it was decided that beyond 18 permutations the generated sentences were devoid of any meaning from which useful conclusions could be drawn.

Some examples of sentences generated by each of the four techniques can be found in Table 6.1.

While our permutations all involve moving or replacing words, we have attempted to keep punctuation relatively unchanged. While the exact procedure varies according to permutation type, this broadly means that when a given word is replaced or moved, any punctuation attached to it will be removed and instead applied to the word now found at the position it used to occupy. We define 'punctuation' for this purpose as any sequence consisting exclusively of following characters at the start of a word – `( ' ` "` – or the following at the end: `. ? ! : , ;) ' ` "`.

Our attempts to avoid altering punctuation are intended to ensure we are genuinely inspecting differences in words, rather than peripheral features of the sentence. Note that we did not consider this goal important enough to devote complicated catch-all systems for comparing punctuation, thus there are numerous edge cases where, for example, one half of a pair of quotation marks will be overwritten by a move. This could arise if the quotation marks were attached to a word which, while not being moved, ends in such a position as to receive the punctuation of a moved word: in this case, the quotation marks

	Reference		At the same time she's a usurper and wants to cook him to death, adds Bajgar.
	Hypothesis		But at the same time, he's an usurper, and he wants to take care of his husband to death, Bajgar said.
	Type	**Degree**	**Result**
1	order	1	At the same she's time a usurper and wants to Cook him to death, adds Bajgar.
2	order	5	At same time and she's a the usurper wants to Cook adds him death Bajgar to.
3	order	15	Usurper Bajgar time to adds death to same and the at a him she's wants Cook.
4	swap	2	At the same time she's a usurper and wants to death him to Cook, adds Bajgar.
5	phrase	3	At she's a usurper and wants to Cook the same time him to death, adds Bajgar.
6	phrase	7	To Cook him to death at she's a usurper the same time and wants, adds Bajgar.
7	choice	3	Of the same husband take an usurper and wants to Cook he to death, adds Bajgar.

Table 6.1: Example sentences produced by different automatic permutation algorithms from a single reference

will be overwritten. While this results in somewhat stilted sentences, we consider that the errors introduced are both rare enough and similar enough to mistakes made by real translation systems that this is unlikely to have any significant effect on our sentences.

In addition to retaining punctuation, we have attempted to normalise the capitalisation of words we move using a very simple truecasing technique. Given the importance of capitalisation for proper nouns and the starts of sentences, we have simply counted the number of times in our reference corpora each word is found with a capital first letter, ignoring all words which begin their sentence. Then, after each permutation algorithm is complete, output sentences have all words' first letters set to uppercase or lowercase depending on whether or not it occurred at least twice as often capitalised than not. Note that in the example in Table 6.1, this has incorrectly resulted in the word 'cook' being capitalised due to its frequent use as a proper noun in the reference corpora.

Flat 'order' permutations

Our 'order' permutations involve simply moving individual words to new positions within the sentence. This permutation type represents simple, entirely chance-based order distortion, with no intelligence whatsoever behind the choice of words to be moved or positions for them to be moved to.

As described in Algorithm 3, this is done by selecting a number of words equal to the current permutation degree, and moving one either left or right by a number of positions equal to the degree, then moving the next word one fewer positions, and so on until the

Algorithm 3 Sentence permutations: 'order'

```
 1: procedure PERMUTE WORD ORDER($sentence, degree$)
 2:     $used \leftarrow \{\}$
 3:     for $distance$ from $degree \rightarrow 1$ do                      ▷ Descending order
 4:         $p \leftarrow$ random position in $sentence$
 5:         while (word at $p$) $\in used$ do
 6:             go to 4                           ▷ Don't move the same word multiple times
 7:         $w \leftarrow$ word at $p$
 8:         $used \leftarrow used \cup \{w\}$
 9:         $d \leftarrow$ either $distance$ or -$distance$      ▷ Choose to move either left or right
10:         Remove $w$ from $sentence$
11:         Insert $w$ into $sentence$ at position $p + d$
12:         if $d > 0$ then                          ▷ If moving right, copy punctuation to
13:             $punctuation$(word at $p + d$) $\leftarrow$ $punctuation$(word at $p + d - 1$)  ▷ new
                                                   ▷ position from word which was there
14:             $punctuation$(word at $p + d - 1$) $\leftarrow \epsilon$      ▷ and remove it from the word
                                                          ▷ which was at new position
15:         else                                                ▷ If moving left, move
16:             $punctuation$(word at $p$) $\leftarrow$ $punctuation$(word at $p + d$)   ▷ punctuation
                                                       ▷ from new position to old and
17:             $punctuation$(word at $p + d$) $\leftarrow \epsilon$          ▷ remove it from new position
18:     return $sentence$
```

last word is moved only one place. We ensure that no word is selected to be moved multiple times during this process, though in practice successive moves may in rare cases cancel each other out.

Simple 'swap' permutations

The simplest permutation type is 'swap', whereby pairs of words have their positions reversed in the sentence. We restrict the pairs to require both words to have the same part of speech, as determined by an external tagger. For simplicity, we have used the Textblob Averaged Perceptron Tagger bundled with NLTK [Bird, 2006] to determine parts of speech. Punctuation, as mentioned earlier, remains in its original position. Similarly, words can not be re-used in multiple pairings: a sentence with five words with the NNP tag, for example, will never swap more than four of them within a single application of the algorithm.

The goal of 'swap' permutations is to highlight the simple confusion of two words during translation. For example, should the subject and object of a sentence be swapped, a reader may have great difficulty in understanding the original meaning, while swapping two adjectives for a single word may have a much lesser effect. These permutations are an attempt to generalise to any part of speech to determine whether there is an overall trend for how much this kind of inaccuracy impairs the sentence's communicative power.

Algorithm 4 Sentence permutations: ‘phrase’

```
procedure PERMUTE PHRASE ORDER($sentence, degree$)
    $parse \leftarrow$ dependency parse of $sentence$    ▷ In practice these variables are only
    $subtrees \leftarrow$ all subtrees in $parse$    ▷ calculated once and cached between calls
    $bounds \leftarrow$ positions of all leftmost nodes in $subtrees$, plus $length(sentence)$
    $subtrees \leftarrow \{s \in subtrees \mid s$ is projective (contiguous), $|s| \leq 9\}$
    choose any $S_{all} \in subtrees$ such that no sets overlap and $\sum_{s \in S_{all}} |s| = degree$
                                                    ▷ Choose $degree$ nodes to move
    for $s \in S$ in descending order of $|s|$ do
        $p_{old} \leftarrow$ position in $sentence$ of leftmost word in $s$
        $p_{new} \leftarrow$ random position in $bounds$ excluding those within any $s' \in S$
                                    ▷ Find the phrase if it’s moved due to prior iterations
        Copy $|s|$ nodes from position $p_{old}$ in $sentence$ to position $p_{new}$
        Remove $|s|$ nodes from $sentence$ at $p_{old}$
                                    ▷ Note that $p_{old}$ may have moved if $p_{new}$ is to its left
        if $p_{new} > p_{old}$ then                          ▷ If moving right,
            $punctuation($word at $p_{new} - 1) \leftarrow punctuation($word at $p_{new} - |s| - 1)$
                              ▷ move punctuation from last position in $s$ to last word in $s$
            $punctuation($word at $p_{old} \leftarrow punctuation($word at $p_{new} - |s|)$
                                    ▷ and from first word in $s$ to word where $s$ was
            $punctuation($word at $p_{new} - |s|) \leftarrow \epsilon$
                            ▷ Punctuation is removed from the words it has been copied from
            $punctuation($word at $p_{new} - |s| - 1) \leftarrow \epsilon$
        else                    ▷ If moving left, copy punctuation from the rightmost
            $punctuation($word at $p_{old}) \leftarrow punctuation($word at $p_{new} + |s| - 1)$
                              ▷ word in $s$ to the word which is now at $s$’s old position, and
            $punctuation($word at $p_{new} + |s| - 1) \leftarrow \epsilon$  ▷ remove it from the end of $s$
    return $sentence$
```

Structured ‘phrase’ permutations

The most complex of our permutations are ‘phrase’ permutations. These are an attempt to take advantage of the same structural knowledge which both DTED and DERP use, to produce slightly more meaningful word movements than our other, simpler algorithms. Described further in Algorithm 4, we detect groups of words within sentences by parsing them into dependency trees and extracting contiguous subtrees of limited sizes from these. We also note down the edges of these subtrees – the transition points between any subtree and its neighbour – to store a set of ‘phrase boundary’ positions. We use the Malt parser with the MaxEnt Treebank part-of-speech tagger for this process.

From this set of component subtrees, for each permutation we select at random a subset containing a number of distinct words which total the current iteration’s permutation degree. For each of these subsets, all component words are then moved as a single unit to one of the predetermined phrase boundaries; this is to ensure that all phrases still make as much sense as possible within the sentence by ensuring they remain as intact as possible, rather than having component words separated by other, moved phrases.

Our motivation for producing ‘phrase’ permutations is to investigate the importance of cohesive phrases within our permuted sentences. This is primarily interesting when the scores given for ‘phrase’ permutations are considered in the context of those for simpler ‘order’ or ‘swap’ outputs: we hypothesise that more words can be moved within phrasal chunks – that is, within relatively semantically coherent units – than can be moved individually while nonetheless retaining the comprehensibility of a sentence.

Word replacements: ‘choice’ permutations

Our final permutation type is not in fact related to word order at all: ‘choice’ refers to replacing words with related ones to assess the relevance of specific word choice. Despite this, we use the term ‘permutation’ for consistency. The intention behind the inclusion of ‘choice’ permutations is to observe the behaviour of tools when word order is not in fact a relevant factor: we expect that order-focused tools tools such as our own will perform relatively poorly at predicting the quality of errors related only to word choice, while those designed for more holistic contexts should by contrast perform normally.

‘choice’ is the only permutation type which requires a machine-translated hypothesis in addition to a human-produced reference translation. As described in Algorithm 5, it involves randomly choosing a number of words equal to the current iteration’s degree, and replacing each with similarly randomly chosen words from the hypothesis translation. Such replacements must have the same part of speech as the word to be replaced, to ensure a degree of relevance to the position in the sentence. Also, to ensure that resulting sentences are different from those of simple word movements, no candidate may already exist within the original or partially-permuted reference translation.

This procedure can have two effects: it may replace a word with a synonym or other word chosen by the automatic translator to fulfill the same purpose, or it may replace it with an unrelated word from elsewhere in the sentence. We consider that both outcomes are interesting.

Our goal in including this permutation type is primarily to verify our participants’ responses, in addition to providing data about choice-based errors for future researchers. We expect scores to indicate a low to non-existent level of order-related incorrectness for these permutations, with anything else strongly suggesting negligence on the part of our judges.

6.3 Survey structure

6.3.1 Medium

In gathering human judgments for WOJ-DB, we needed to decide what format of questionnaire to provide to participants. The two primary alternatives were to use an online survey system or an offline paper-based one. While both media have powerful advantages, we ultimately chose the latter for several reasons.

Before describing these, we note that online survey systems such as Amazon Mechanical Turk [Amazon, 2005] are able to boast diverse user bases [Paolacci *et al.*, 2010] and low costs [Quinn and Bederson, 2011], allowing a large number of high-quality judgments to be gathered with a minimum of effort once a project has been set up.

Algorithm 5 Sentence permutations: ‘choice’

```
 1: procedure PERMUTE WORD CHOICE($reference, hypothesis, degree$)
 2:     $output \leftarrow reference$
 3:     repeat
 4:         $w \leftarrow$ random word in $output$
 5:         $w' \leftarrow$ random word in $hypothesis$
 6:         if $w'$ has a different part of speech from $w$          ▷ Match parts of speech,
 7:             or $w' \in reference$ or $w' \in output$ then         ▷ and don’t re-use words
 8:             go to 4 or 5, or exit     ▷ Go to 5 if only this $w'$ is invalid, 4 if no valid $w'$
                    ▷ exists for $w$, or exit (producing nothing) if no $w$ exists with a valid $w'$.
 9:         $punctuation(w') \leftarrow punctuation(w)$               ▷ Transfer punctuation from
                                                     ▷ reference word to hypothesis word
10:         Replace $w$ with $w'$ in $output$ ▷ Replace reference word with hypothesis word
11:     until $degree$ words have been replaced
12:     return $output$
```

Conversely, the benefits of a paper system include a much greater understanding of the backgrounds of the judges we would recruit, more of an accountability trail should any information prove faulty for whatever reason, and a much simpler process of producing and distributing questionnaires along with payments.

This last reason of simplicity was primarily why we selected a paper survey for our pilot study (Section 6.4). As described in that section, our pilot provided evidence that the numbers of participants we would require for our main survey would be within that permitted by our financial resources, in addition to being low enough that local participants would still provide reasonable levels of demographic diversity. We thus chose to conduct our main survey in paper form, by recruiting individuals locally as described in Section 6.5.

When recruited, for ethical reasons participants were provided with a number of documents in addition to the main survey. First was a briefing sheet, containing essential information about the study while being nonspecific enough to avoid introducing bias. This is reproduced in Appendix A, pages 162-163. They were also asked to sign a Participant Consent form, agreeing for the information to be used in various ways as shown in Appendix A on pages 164-165. After this they were presented with the main survey, on completion of which they were given a further Debriefing information sheet, providing a little more information about the goals of the study and reproduced in Appendix A on page 166.

These information and consent sheets were written using traditional word processors, but to generate the main questionnaires we used an automatic system named SDAPS [Berg, 2015] upon a LaTeX base. The combination of these tools provided a number of useful features. First, unlike more interface-focused word processors, LaTeX’s compilation procedure allows source files to be generated automatically using shell scripts. These files would thus be consistently and elegantly formatted to seamlessly provide a final document without any user input being required. These benefits were important when generating dozens of surveys – both for saving time and, more importantly, for ensuring consistency across surveys by eliminating per-survey human error.

While LaTeX on its own does not provide extensive functionality specifically designed for surveys, SDAPS provides various tools which allowed professional-level formatting with a minimum of effort. Thus, the formatting for Likert scales, separate sections and user instructions were all dealt with by its interface.

It was once the surveys were generated that SDAPS became most relevant. One of its core features includes scanning of completed PDFs, using barcodes and formatting to extract as much information as possible for processing. This allowed the hundreds of individual scores to be gathered automatically and put in a format which could be easily passed to post-processing tools, again eliminating a significant source of human error. Of course, text fields such as participants' ages could not be processed by this tool, and were manually typed according to bespoke scripts after scores had been gathered.

In terms of contents, the survey consisted of two parts: an overview sheet identical for all participants, and the main bulk of the questionnaire with varying sentences. These were presented as a single document. The overview sheet contained questions about participants' backgrounds and motivations, along with a single sample sentence with two-line explanations of the questions the participants were to be asked. This page may be seen in Appendix A, page 167.

6.3.2 Participant information

The background information we required from participants was primarily to include any information possible about their level of experience with English and other languages. We also endeavoured to avoid any questions which would allow participants to be identified: beyond their name and signature on the Participant Consent Form (Appendix A, pages 164-165), participation was entirely anonymous.

The simplest questions were not directly related to language, but simply focused on the demographics to which the given participant belonged. We did not expect differences in responses to these questions to have any statistically significant bearing on score trends, but considered such variation possible enough that the data was nonetheless worth collecting in order to verify our expectation. The demographic factors we asked about were as follows: the participant's age in years; their gender; and the nationality they felt best represented them.

We also asked for participants' education level, hypothesising that those with higher-level degrees might approach language with more precision than those without. Another area more likely to have significant effects on participants' response trends was whether or not they had disabilities such as dyslexia; we asked them to list any which they felt might impact their reading.

For specific linguistic knowledge, the most important question was their level of experience of English. We asked this in two ways: first, asking whether English was their first language; and second, asking them to evaluate their fluency in it along with others. The former question was primarily intended for non-native speakers, asking how long they considered they had been speaking English fluently to gain a limited understanding of their investment in the language.

The second question, about participants' general fluency level in any languages, was intended primarily to explore secondary languages. Participants were asked to include their self-rated fluency level along with any qualifications they had achieved in them.

English was explicitly included in this question primarily to suggest those with advanced qualifications in English, such as University degrees, to indicate this. While we do not in our statistical analysis inspect the qualifications each participant has in any language, the information was considered valuable context for future researchers using the database.

Finally, to provide a less objective measure of participants' attitude to language, and thus potentially one which would better indicate their strength of opinion, we asked how much they noticed grammar and sentence structure in everyday life. For this question, they were given a five-point Likert scale ranging from "It doesn't affect me at all" to "Correct use of language is important". This was included both for future researchers, in case attitude is especially relevant to other projects, and simply to permit tests within our own of whether such subjective factors might contribute to their response styles.

6.3.3 Sample sentence

The sample sentence, included immediately after the Participant Information questions on the survey's cover sheet, was included in order to familiarise users with the format of the rest of the survey, and to provide them with an explicit opportunity to decide for themselves the extent to which different errors affect the overall quality of the sentence.

The sentence itself was neither generated by translation systems nor by automatic permutation, but was written manually. This was with a view to including several types of relatively clear errors in a situation where the correct meaning was nonetheless not wholly lost. Participants were explicitly told that their answers to that sample question would not be used for research purposes, and as such all results found in Section 7 omit this sentence.

While we considered that at least one sentence by which participants could calibrate their own assessments of various errors' significance was important, it would have been possible to include several of these. This would have provided a chance to homogenise scoring to a certain extent across judges, had such sentences included several edge cases: sentences with only ordering errors, sentences with many trivial errors which nonetheless clearly retained their meaning, and so on.

However, given that our primary use for such sentences was simply to provide an opportunity for participants to change their minds or ask questions, something achieved by a single sentence, we chose to include just one. This allowed us to keep the survey length shorter so as not to overload our participants with over 51 sentence pairs, while also not compromising the number of usable datapoints by having a greater proportion of our sentences be such samples.

6.3.4 Sentence selection

After the cover sheet, participants were presented with 50 pairs of hypothesis and reference sentences, each with the two questions shown in Section 6.2.1. These sentences were accompanied with the name of the language from which the sentences were translated, but no further context.

Exactly half of these sentences were taken from real hypothesis translations submitted to WMT, as described in Section 6.1.1. These were taken entirely at random from the pool of all sentences which had been used as a basis for automatic permutations as described

Year	No. corpora	Min. length	Max. length	Total length	Permutations
2014	35	789	1339	35011	74186
2015	53	497	909	37626	45588
2016	56	1999	3000	160951	93176
Total	144	497	3000	233588	212950

Table 6.2: The sizes of various WMT corpora included in the survey. Each corpus represents all sentences submitted to WMT by a given system, e.g. Edinburgh's Russian-English phrase-based system at WMT 2014 [Durrani *et al.*, 2014]. The 'length' of a corpus is the number of sets of source, reference and hypothesis sentences it contains.

in Section 6.2.3 and Table 6.2: thus, 450 sentences from each corpus within the WMT corpora for 2014, 2015 and 2016. No restrictions were placed on the number of sentences within a given survey which could come from a specific year, translation system or source language.

The other half of the survey was composed of automatic permutations, including at least four produced by each algorithm with the remaining such sentences being of randomly selected types. The permutations were all based on 'real' hypothesis sentences which existed in the same survey, with one or two permutations being included for each of several such hypotheses. Thus, for the majority of permuted sentences a given judge is presented with, they will elsewhere in the survey also be presented with another permuted sentence of the same type but with a different degree, and also with a genuine machine translation associated with that reference.

Within both the permuted sentences and genuine hypothesis sentences, we have ensured that a significant minority of those included in each survey are included in every other survey too: specifically, 40% of each group, making 40% of each complete survey. Referred to as 'shared' sentences, these were preselected both to ensure that each survey included a range of qualities and error types, and to permit statistical comparisons of different judges' scoring trends, as described in Sections 6.2.1 and 6.2.2.

The exact ratios of sentences which, for any given survey, are taken from each different pool are shown in Table 6.3.

6.3.5 Feature distribution

While the above procedure for selecting sentences contains a number of restrictions intended to result in equitable distributions of hypothesis sentences and automatic permutations, it nonetheless relies on a significant level of random selection. We must thus inspect the distribution of key features of the sentences included in the survey. We consider that these features to be fourfold: sentence type (i.e. machine-translated hypothesis or automatic permutation), sentence length, sentence quality and source language.

Of these factors, sentence type may be the simplest to inspect. The numbers of each type of sentence must, as described above, be equal in each questionnaire. We further required that this evenness should apply also to both the sets of sentences which are unique within each survey and those which are shared among all surveys.

No. of sentences	Sentence type	Permutation type	Same hypothesis	Shared
10	hypotheses			✓
2	permutations	order	✓	✓
2	permutations	swap	✓	✓
2	permutations	phrase	✓	✓
2	permutations	choice	✓	✓
2	permutations	*random*	✗	✓
15	hypotheses			✗
2	permutations	order	✓	✗
2	permutations	swap	✓	✗
2	permutations	phrase	✓	✗
2	permutations	choice	✓	✗
7	permutations	*random*	✗	✗

Table 6.3: Breakdown of sentences in each questionnaire. All groups marked as having the same hypothesis contain two permutations based on the same sentence, with different base hypotheses for each row in the table. 'Shared' sentence pairs are the same across all questionnaires.

We investigate the distribution of sentence type together with that of sentence quality. Two measures of this were available to us during the generation of WOJ-DB, while two additional sets of scores were of course obtained during the survey.

The measures available while creating WOJ-DB were dependent on the type of sentence. The majority of sentences extracted from WMT had ranking information, representing their adequacy as judged by volunteers at the conferences, with each sentence measured relative to a small group of up to four others as described in Section 6.1.1. 29.1% of unique sentences did not have any WMT rank assigned by judges, an unfortunate unnecessary consequence of the random selection of sentences. Recall that we converted this rank, when available, into an absolute score using the somewhat simplistic process described in Equation 6.1.

While the quality of hypothesis sentences from WMT are ranked according to the ranks at the workshops, no human assessments of automatic permutations existed before running our survey. We have thus used the permutation degree (Section 6.2.3), the number of words altered by the permutation algorithm. We normalise this in a similar manner to WMT hypotheses, according to the maximum permutation of any sentence in the entire database $sents$:

$$score_{perm} = 1 - \frac{degree(perm) - 1}{max(degree(p \in sents)) - 1} \; \forall perm \in sents \tag{6.2}$$

The two measures of sentence quality produced during the survey were that of holistic and order-focused adequacy, as per Sections 6.2.1 and 3.3.2. While in the survey participants were provided only with labels for the extreme ends of the scoring scale, as described in Section 6.2, their five possible scores have for our purposes been converted into numerical values in a similar manner to the above. Thus, for example a score of 0.25 indicates a participant having selected the second box from the left (worst) for a given question.

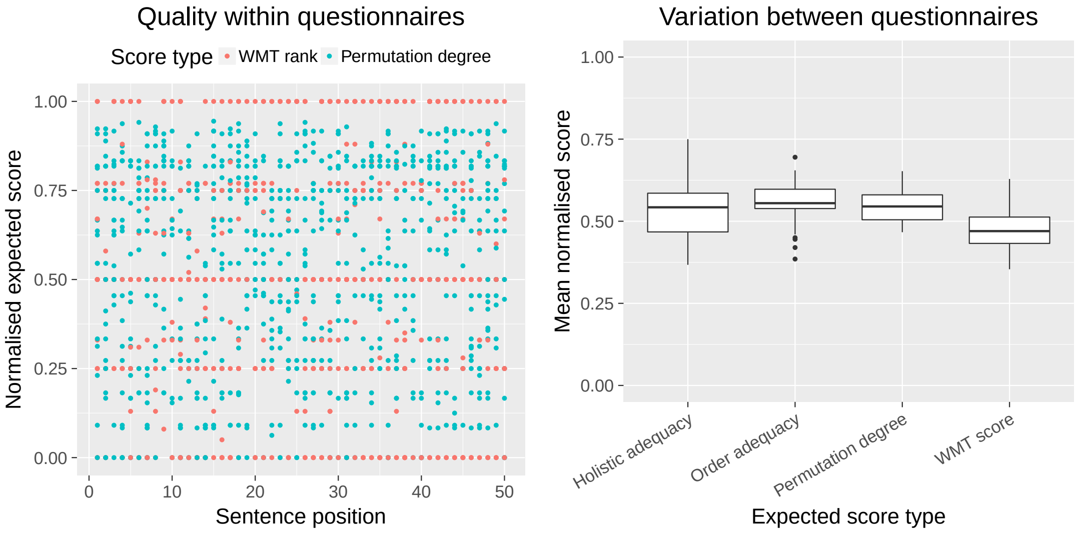


Figure 6.1: Quality distribution between questionnaires, based on sentence type, WMT rank, permutation degree or survey scores; and within questionnaires by position and sentence type

Figure 6.1 contains information about how each of these quality judgments varied across different questionnaires and different positions within each questionnaire. The left-hand graph indicates where in any survey sentences of a given quality are found: we can see that sentence quality is broadly evenly distributed throughout the questionnaires, as we would expect from the random distribution mechanism we used when generating them.

The right-hand graph in Figure 6.1 shows the variation in quality between different questionnaires. The mean value for each type of quality has been calculated for each survey and plotted. For true uniformity we would expect mean values of 0.5 in all cases; in practice we see distributions very similar to this for all four score types.

Through Figure 6.2, we can inspect the distribution of another essential feature of our sentences: their length. Both graphs indicate the distributions of lengths: in the left-hand one we can observe the lengths of source, reference and hypothesis sentences throughout the survey. We can see a very approximate normal distribution of lengths, with a mean of just under 20 words for all three sentence types.

In the right-hand graph we can see that these lengths vary in a tight linear relationship, with a Pearson's correlation coefficient of 0.95 between reference and hypothesis sentences as indicated by the red regression line: participants were very rarely faced with pairs of sentences with dramatically different lengths from each other.

Finally, Figure 6.3 shows the distribution of source languages for sentence pairs across all surveys. While sentences translated from German and Hindi are the most common, and Turkish by far the least prevalent, the differences are not vast: even German and Hindi are only approximately twice as common as the second-least-common language, Romanian.

We thus consider that participants were presented with sentences translated from an appropriate range of source languages, providing variety in our translations and avoiding

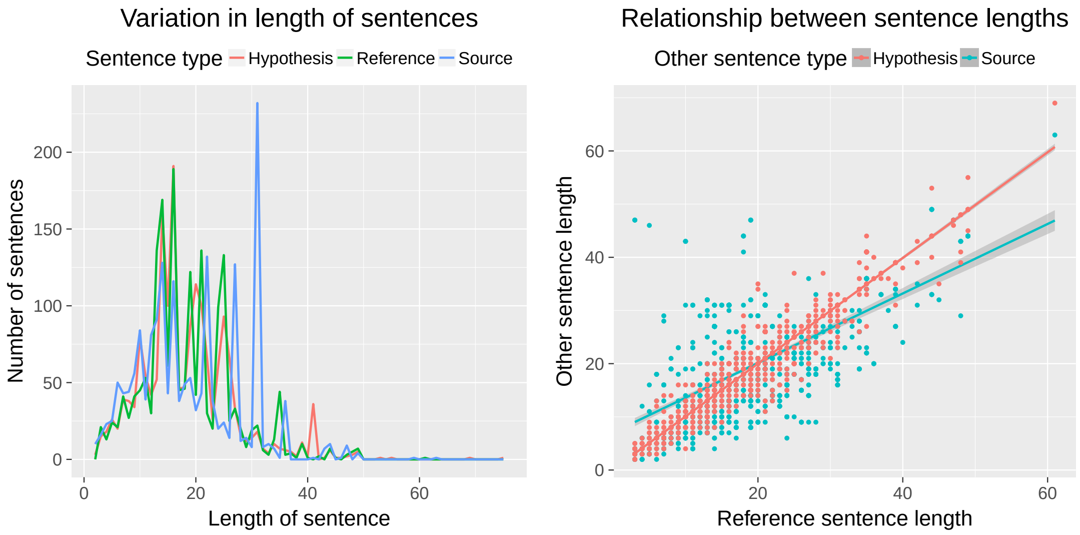


Figure 6.2: Sentence length variation within all sentences used in WOJ-DB

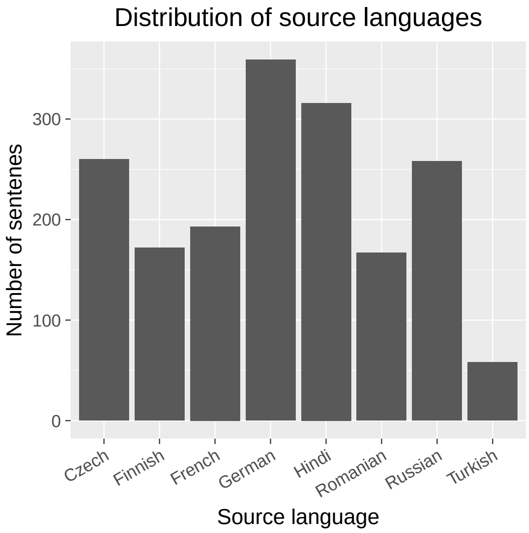


Figure 6.3: Source languages for sentences in WOJ-DB

potential pitfalls which might have arisen if all sentences were translated from the same source. This allows results based on such sentences to be used in as broad a range of linguistic contexts as possible, as per our design goal of Flexibility.

6.4 Pilot study

6.4.1 Execution

Prior to producing the real survey for WOJ-DB, we ran a pilot study. This small-scale experiment had several purposes: to obtain feedback about any aspects of the survey which could be considered unintuitive; to gain an approximate view of how long the completion of any given survey would take; and to provide an estimate of how many surveys would need to be completed in order to achieve statistical significance with our results.

Our pilot study was performed on paper, for the reasons described above: we wished to test whether restricting to a small group would create unacceptable limitations in available participants, which could have forced us to go online despite our reservations. Our participant pool for the pilot study was five individuals, approached personally within the University of St Andrews. No financial incentive was offered to any of the pilot participants, and no individual took part in both the pilot and the main study.

The sentences used in the pilot study were almost the same as those included in the main study, with a few changes within the automatically permuted sentences. While ratios and generation techniques remained as described earlier for almost all sentences, 'phrase' permutations were not present; instead, a separate type of permutation named 'both' was used. The algorithm for this was very simple: for any given sentence and degree, first the algorithm for 'choice' permutations was run, then that for 'order' permutations on the result.

Figure 6.4 shows basic information about the demographics into which the pilot participants fell.

6.4.2 Results & resultant changes

Feedback from the pilot was limited in quantity but varied. A comment was made that simply providing the gender options 'Male', 'Female' and 'Prefer not to say' was potentially discriminatory to those who identified in a less binary way. A number of comments were made on individual sentences, but broadly the format of the questionnaire was received positively.

In the pilot study, the 'correct' human-produced translations were shown first, followed by the relevant machine-generated sentence. Participants commented that this precluded a genuine attempt to guess at the meaning of the hypothesis sentence before discovering its true meaning, and suggested that being aware of the intended meaning before reading the less reliable sentence could have produced a significant bias towards assuming the latter was more meaningful than it would have appeared on its own. In response to this comment, the order of the two sentences was inverted.

Since the pilot study, we also introduced 'phrase' permutations and discontinued 'both' permutations. The former was created because within the pilot study we felt that the existing permutation techniques did not use any of the information and intelligence of which we tried to take advantage within our tools DTED and DERP, and as such the insights they could provide on different error types was not adequate for our purposes.

We removed 'both' permutations for two reasons. First, it was not considered that they represented any genuinely novel feature within sentence quality which could not be better understood by inspecting the judgments on either 'choice' or 'order' permutations. Second, with only 25 permuted sentences of any kind within each survey, we wished to minimise the number of different categories they needed to be split into.

In pursuit of an estimate of how many surveys would need to be run in our final study if we were to attain statistical significance with our results, we have calculated statistical

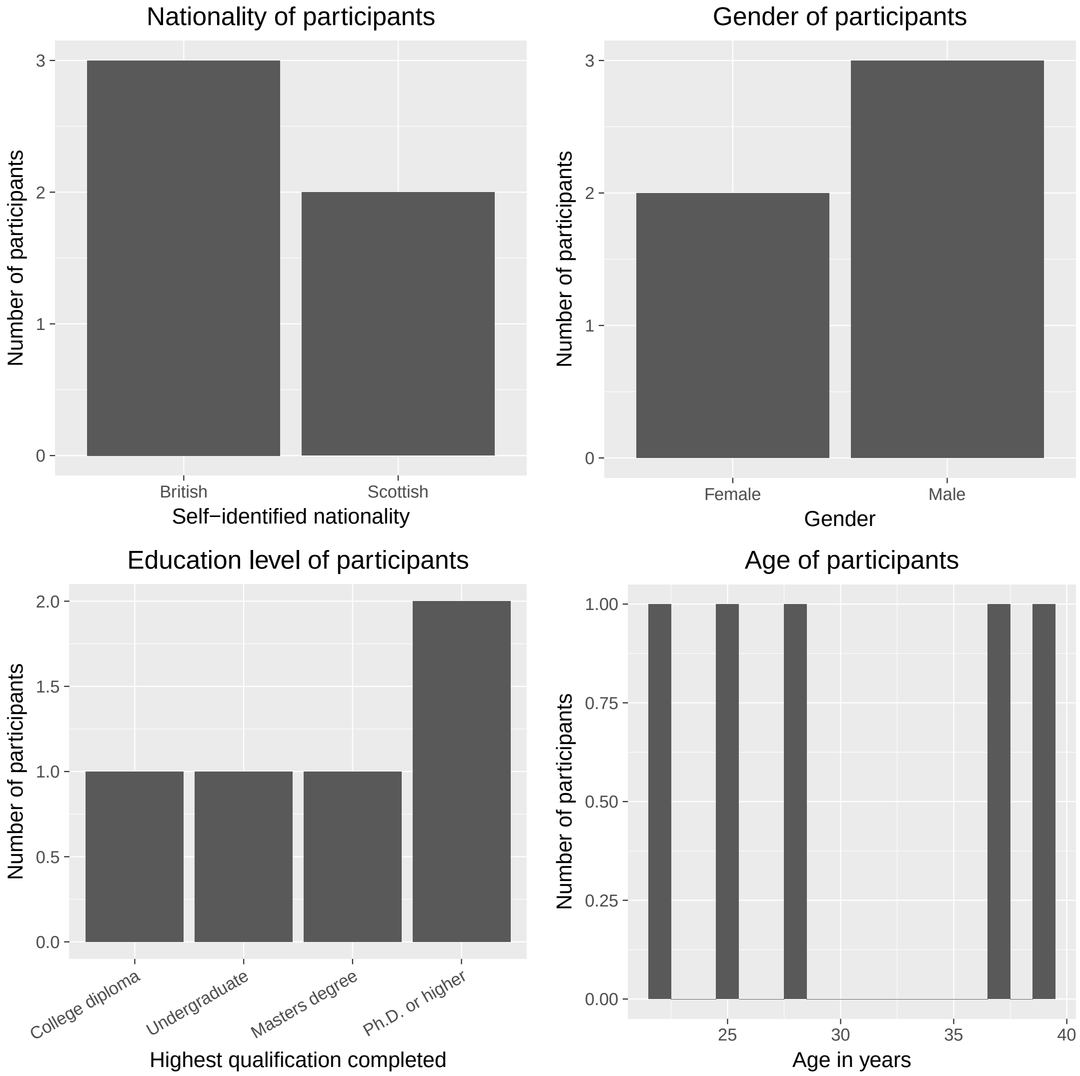


Figure 6.4: Basic demographic details about pilot survey participants

power based on our pilot study using the technique described by Chow *et al*. [2008, p. 71]. While our intention (Chapter 7) is to inspect correlations calculated using the scores gathered in WOJ-DB, for our statistical power calculations we have modelled a simpler case: simply calculating pairwise differences in means between scoresets through a two-tailed one-way Analysis of Variance or ANOVA test.

In our calculations, we have compared the results to each of our two survey questions with scores for each variant of DTED and DERP. Most (82.5%) of the results of these calculations suggested that scores for fewer than 100 sentences would be sufficient to attain statistical significance, while 5% indicated that 1000 would be enough and 12.5% expected dramatically more than 1000 datapoints to be necessary.

Our conclusions, based on these calculations, were that while the 100 datapoints contained within a mere two surveys could potentially be adequate for our needs, our experiments – and any others which might in the future be run on our dataset – would benefit from as many datapoints as reasonably possible. We thus recruited 30 paid participants and another 6 voluntary ones, considering this to represent an appropriately large base without requiring unreasonable financial investment.

6.5 Participation

Once our survey was designed, prepared and piloted, it thus needed to be given to as wide a participant pool as possible. There were a number of ways through which we recruited participants, who ultimately came from a range of different demographic groups.

6.5.1 Recruitment

University Memos

Our primary method of advertising the study was by 'Wednesday Memos', a system provided by the University of St Andrews whereby individuals are each provided with a weekly list of announcements. These lists are prepared separately for undergraduate students, postgraduate students and staff, but with no more personalisation than that. In order to attract as wide a range of individuals as possible, we requested that our advertisement be sent to all three groups.

Our intention with such a broad advertisement was to give as little information as possible, to minimise self-selection biases, while ensuring participants knew all relevant practical details. Nonetheless we felt it necessary to include at least a general domain area, as not doing so would be unorthodox for such memos and might, as such, put off a number of readers. We mentioned the financial incentive available in order to provide an obvious reason to participate other than the subject matter.

The three memos were identically worded (and included a minor accidental punctuation error), as follows:

> **Participants wanted: native English speakers for language survey**
> Are you a native English speaker? Would you like a £5 Amazon voucher? Participants are needed for a one-hour study on automatically produced translations within the School of Computer Science North Haugh). For more information contact Martin McCaffery at mm689@st-andrews.ac.uk. (Ethical approval: CS12370)

When interested parties got in touch by email, they were informed that the study would likely take somewhat less than an hour, but were given little more information beyond that which was needed to organise a time to meet in person. They then arrived and were given the various information sheets and the survey; after completion of all of these, no further communication was made. Note that the vast majority of participants took significantly less than the stated hour to complete the survey, with most taking approximately 30-40 minutes.

Using the Memos system, we were technically able to reach the entire body of approximately 10,660 students [Higher Education Statistics Agency, 2016] and 2,485 staff members [University of St Andrews, 2015] (approximated using data published for the academic year 2014/15). In practice, many may not have received or read the memos for a variety of reasons, but we have no means of estimating the number of such cases. Of

Group	Advertised to	Responded	Participated
University members (via Memos)	13145 (approx.)	29	18
St Andrews Quaker Meeting	30	6	5
University of St Andrews Jujitsu Club	6	6	5
Other individuals	8	8	8

Table 6.4: Breakdown of the various groups of whom members participated in the survey

those who did read the memos, 29 individuals expressed an interest, of whom 18 actually participated in the survey.

Word of Mouth

In addition to the University's Memos system, a number of participants were recruited by simple word of mouth. This was in order both to increase the quantity of judges and judgments we received, and to broaden the scope of our participants beyond those University members who were likely to read the memos they received. Only one group was truly unrelated to the University, while the personal interaction with the others was in addition to, rather than instead of, them.

Several groups were approached, fitting into three broad categories. First, the St Andrews Quaker Meeting is a religious group, unrelated to the University and primarily composed of individuals aged over 30. Next, the University Jujitsu Club is a martial arts group affiliated to the University. Very few members of the former group would have received the University memos, while all of the latter group would have done. Finally, a small number of individuals were approached personally about their participation; these were all members of the University. The numbers within each group can be found in Table 6.4.

6.5.2 Demographics

While the original memos publicised to the entire University restricted participants to being native English speakers, no other communication forms made this restriction. This was because while we expected the majority of our participants to be found within the University, and thus wanted to restrict that pool to those who would have the most reliable insights into the English language, we nonetheless wished to have a number of judges for whom English was not their native language for the sake of flexibility, as described in Section 6.2.1. In the end, we had two participants for whom English was not their native language, with participant IDs of 17 and 38.

Similarly, while no explicit restrictions were made on the grounds of disability, we had only a single judge who declared any form of disability which might have affected their reading ability; this participant, with ID 32, has dyslexia.

Our participants ranged in age from 18 to 74, with the vast majority being at the younger end of this range. The vast majority considered themselves simply British, while a handful considered themselves specifically Scottish. We had five more female participants than males. Details on all three of these aspects can be found in Figure 6.5.

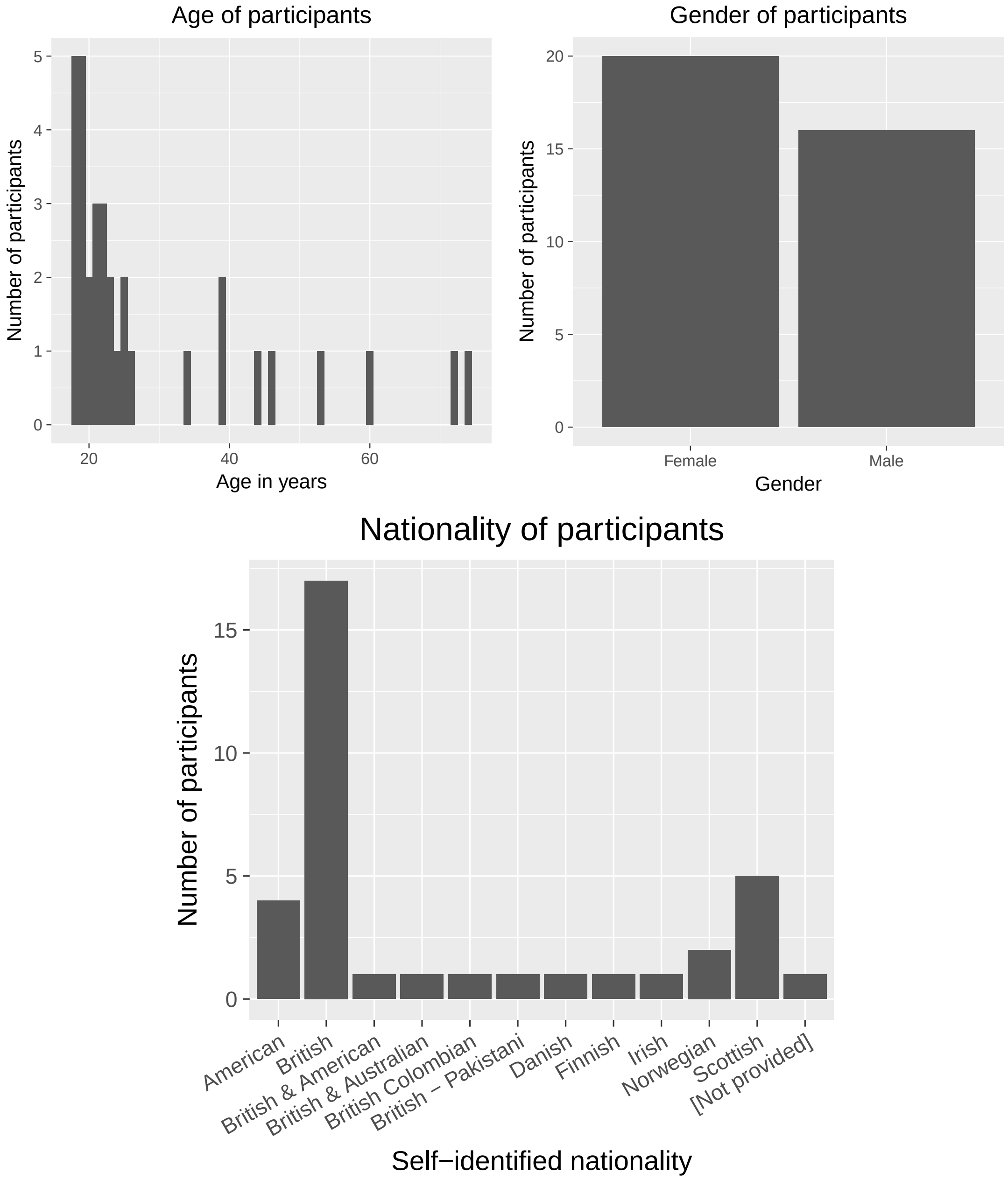


Figure 6.5: Basic demographic details about participants

Similarly to participants' ages, their education levels ranged quite significantly, though the vast majority had completed either no formal qualifications beyond school level or had a college diploma. Irrespective of education level, however, all participants indicated that they paid at least a moderate level of attention to grammatical accuracy in everyday life. This latter fact suggests a not-insignificant bias, despite our attempts to avoid such, for individuals to take part in our survey only if they were actively interested in language. Both these distributions can be seen in Figure 6.6.

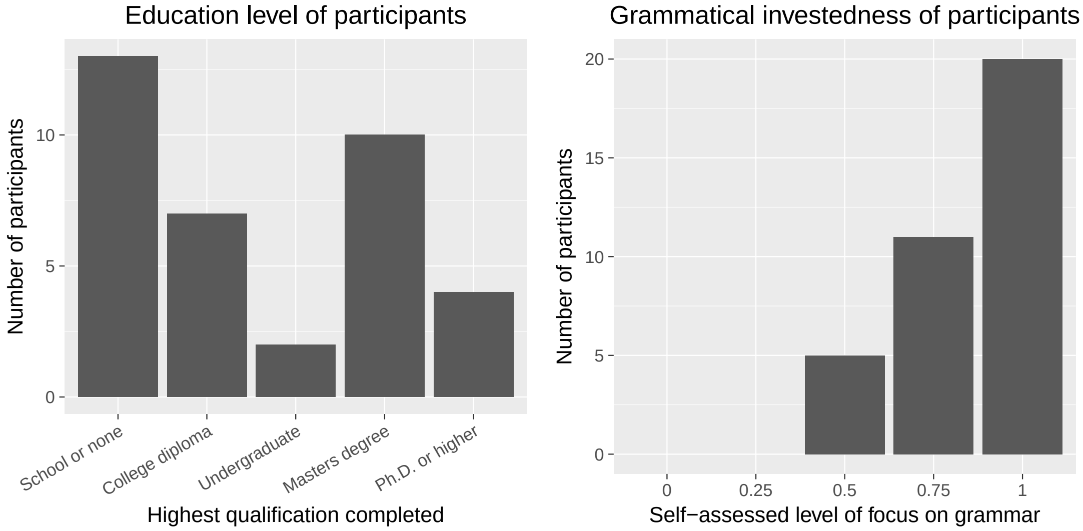


Figure 6.6: Education level and grammatical focus of participants

The numbers of languages other than English which were spoken by participants varied quite dramatically. The distribution of total numbers of languages spoken by participants can be found in Figure 6.7, alongside an aggregation of the cross-linguistic fluency level of participants. To represent the latter, we have combined the number of languages spoken by individual participants with the degree of fluency they have in each. In this way, a participant speaking native English, fluent French and a few words of German would be assigned 7, 4 and 1 'points' respectively for each, giving them a combined total of 12 points.

6.6 Processing & interpretation

Full details of the scores submitted, the variations between these scores based on factors to do with individual judges or sentences, and the relationships between the scores and those of automatic metrics – our own, DTED and DERP, plus several other existing tools – can be found in Chapter 7.

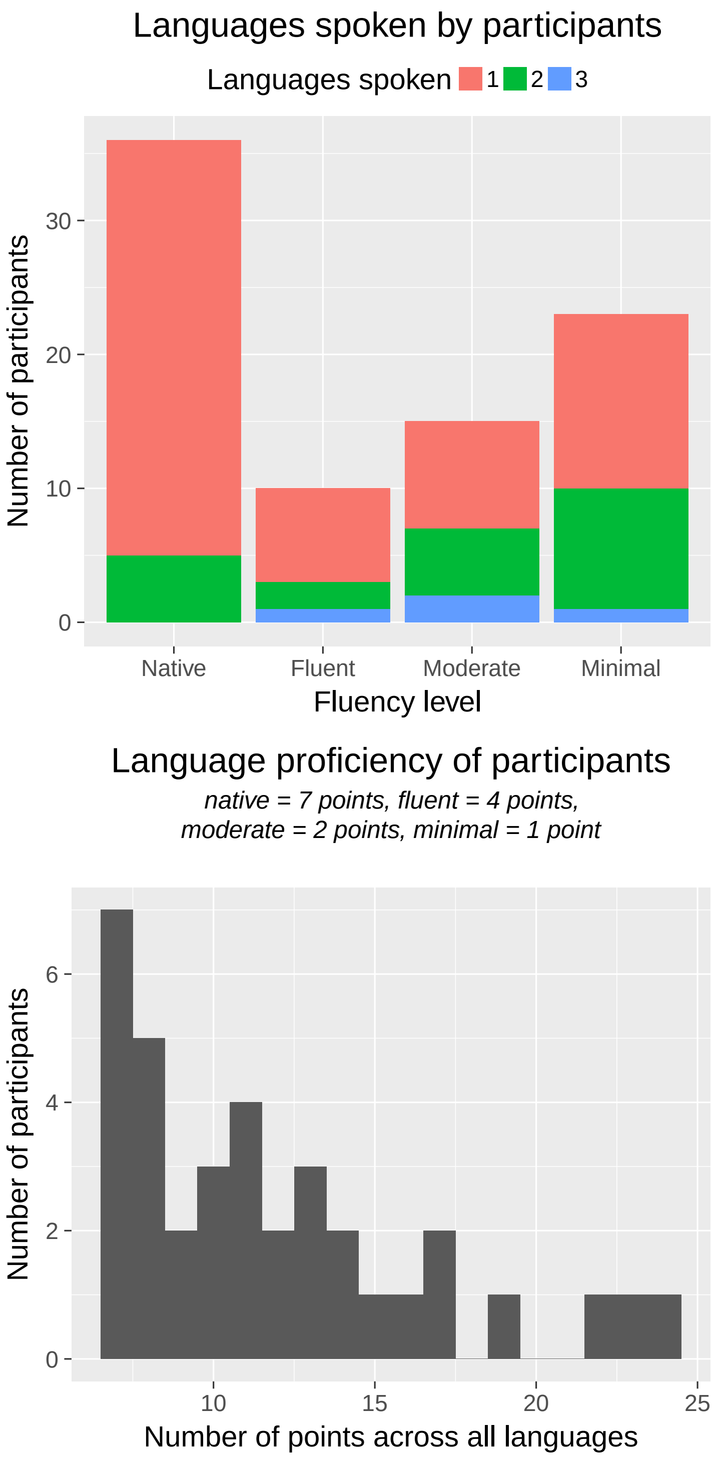


Figure 6.7: Language knowledge of all 36 participants

Chapter Seven

Results & Analysis

7.1 Introduction

In previous chapters, we have discussed the two tools we have produced and the dataset which we have created. DTED (Chapter 4) and DERP (Chapter 5) are intended to represent different, novel approaches to the evaluation of the effect of order in machine translation, while WOJ-DB (Chapter 6) was designed to contain true human-produced assessments of that effect.

Our task now is to evaluate the success of each of these projects, in the context both of each other and of other tools with similar intentions designed by the Machine Translation community. Given the overarching questions proposed in Section 3.2 and the tools we have produced, we attempt to respond to the following:

1. Is WOJ-DB a relevant source of human judgments?
2. Can DTED predict human judgments on word order better than comparable tools?
3. Can DERP predict human judgments on word order better than comparable tools?
4. Is DERP better than DTED at predicting human judgments on word order?
5. Does the inclusion of structure in either DTED or DERP result in a significant increase in their ability to predict human judgments?
6. Can all our tools predict human judgments on order at least as well as human judgments of overall adequacy?

To this end, we perform two broad types of analyses. First (Section 7.2) we use the data within WOJ-DB, along with various external datasets such as sentence ranking data submitted to WMT, to evaluate their relevance as a repository of accurate human judgments. This corresponds to question 1 above.

Our second analysis (Section 7.3) is of our two tools, DTED and DERP, inspecting features related to questions 2 to 6. Using the information gathered in WOJ-DB, we assess the efficacy of DTED and DERP, along with third-party tools such as BLEU and Meteor.

Within these analyses, we attempt to respond to each of the primary research questions described in Section 3.2. Questions 2 and 3 relate to Section 3.2.1; question 5 is based on Section 3.2.2; and question 6 is analogous to Section 3.2.3.

Having investigated the empirical features of each of our projects, we return in Section 7.4 to the questions above. We assess the relevance and meaning of the features we have inspected, and attempt to place our work once again within the broader academic context.

7.1.1 General hypotheses

In order to scientifically investigate the questions we have put forward, we must produce falsifiable hypotheses. We begin by presenting our three primary investigative goals – the accurate evaluation of word ordering, the relevance of structure in doing so, and the relationship between order evaluation and holistic evaluation – as three key pairs of hypotheses.

H_{A}^{EVAL} At least one of the tools we have produced can predict human judgments on word order adequacy to a greater degree of reliability than all community-produced or statistical alternatives we compare them with

H_{0}^{EVAL} No tool we have produced can can predict human judgments on word order adequacy to a greater degree of reliability than all community-produced or statistical alternatives we compare them with

H_{A}^{STRUC} At least one variant of DTED or DERP is more accurate at predicting human judgments on word order adequacy when including dependency structures than when such information is omitted

H_{0}^{STRUC} No variant of DTED or DERP is more accurate at predicting human judgments when including dependency structures than when such information is omitted

H_{A}^{ORD} At least one variant of DTED or DERP is at least as accurate at predicting human judgments on holistic adequacy as on word order adequacy specifically

H_{0}^{ORD} No variant of DTED or DERP is at least as accurate at predicting human judgments on holistic adequacy as on word order adequacy specifically

In order for the above hypotheses to be unambiguous, we must formally define our comparison techniques between tools. When performing our evaluations, we consider that a tool T has a 'greater degree of reliability' than some other T' if *every* available configuration of third-party tools (e.g. dependency parser or alignment generator) used by T results in a better match with human judgments than *any* such configuration of T'. Similarly, T is 'at least as accurate' as T' if the two sets of judgments overlap: if the highest score for each is greater than the lowest score for the other.

We must also specify the exact dataset we will use to respond to each of the above hypotheses. Firstly, we consider all judgments provided by every participant to WOJ-DB. As explained in Section 7.3.5, although we have gathered information from judges belonging to diverse demographics, we consider conclusions based on all available scores to be the most reliable and the most relevant.

Secondly, we base our conclusions exclusively on human judgments gathered specifically for WOJ-DB: no investigation is made of those submitted to WMT or any other venue discussed in Section 6.1.1.

Finally, we apply each hypothesis separately to machine-translated hypothesis sentences and to those generated through automatic permutations as per Section 6.2.3. This is due to the inherent linguistic and structural differences between the former and each type of the latter, which prevent them from being used interchangeably. We discuss the performance of all tools relative to automatic permutations in Section 7.3.3 before discussing conclusions relating to them in Section 7.4.2.

7.1.2 Specific hypotheses

While H^{EVAL}, H^{STRUC} and H^{ORD} represent the most important and general aspects of the experiments we perform, we do not consider that responses to them alone will truly provide a thorough understanding of the phenomena we are investigating. To respond to this, we present a number of more specific hypotheses, as follows.

$H^{EVAL}_{DTED_{\mathrm{A}}}$ One or more variants of DTED have a higher correlation with WOJ-DB judgments on word order adequacy than that of every other investigated tool excluding DERP

$H^{EVAL}_{DTED_0}$ No variant of DTED has a higher correlation with WOJ-DB judgments on word order adequacy than that of every other investigated tool excluding DERP

$H^{EVAL}_{DERP_{\mathrm{A}}}$ One or both variants of DERP have a higher correlation with WOJ-DB judgments on word order adequacy than that of every other investigated tool excluding DTED

$H^{EVAL}_{DERP_0}$ Neither variant of DERP has a higher correlation with WOJ-DB judgments on word order adequacy than that of every other investigated tool excluding DTED

$H^{EVAL}_{DD_{\mathrm{A}}}$ One or both variants of DERP have a higher correlation with WOJ-DB judgments on word order adequacy than that of every variant of DTED

$H^{EVAL}_{DD_0}$ Neither variant of DERP has a higher correlation with WOJ-DB judgments on word order adequacy than that of every variant of DTED

$H^{EVAL}_{CONF_{\mathrm{A}}}$ Correlation between human judgments on word order adequacy and variants of DTED and DERP is affected by no more than 0.122 by the choice of configuration used.

$H^{EVAL}_{CONF_0}$ Correlation between human judgments on word order adequacy and variants of DTED and DERP is affected by at least 0.122 by the choice of configuration used.

$H^{STRUC}_{ALL_{\mathrm{A}}}$ All un-flattened variants of either DTED or DERP have a higher correlation with human judgments on word order adequacy than their corresponding flattened variant

$H^{STRUC}_{ALL_0}$ Not all un-flattened variants of either DTED or DERP have a higher correlation with human judgments on word order adequacy than their corresponding flattened variant

$H^{ORD}_{ALL_{\mathrm{A}}}$ Every variant of DTED and DERP is at least as accurate at predicting human judgments on holistic adequacy as on word order adequacy specifically

$H^{ORD}_{ALL_0}$ Not all variants of DTED and DERP is at least as accurate at predicting human judgments on holistic adequacy as on word order adequacy specifically

We have split H^{EVAL} into two, with H^{EVAL}_{DTED} and H^{EVAL}_{DERP} separately querying the success of our two tools. Further, as DERP is intended to provide more detail and thus a more thorough evaluation process than DTED (see Sections 4.7 and 5.1), H^{EVAL}_{DD} asks whether this deeper analysis results in higher performance in practice.

Additionally, while we rely on third-party systems such as parsers and aligners in our tool, we expect their impact on the functionality of our tools to be slight. While the dependency parse and alignment relation are key information we use, we anticipate that any legitimate parse, as produced by any publishable-quality parser, will result in approximately the same evaluation of our sentences – as encoded in H^{EVAL}_{CONF}.

We have chosen the threshold of 0.122 for hypothesis pair H^{EVAL}_{CONF} as a function of our dataset: a standard deviation. To provide as meaningful a cutoff as possible we have calculated that of *all* correlations related to variants of DTED and DERP, with the exception that we have omitted flattened data due to their more limited use of configurations and greater variability. Note that throughout this chapter, we use ‘configuration’ to refer to the set of these third-party tools used by DTED or DERP, as opposed to a ‘variant’ which refers to the set of flags applied to any execution.

While H^{STRUC} provides a simple measure of whether structure is at all relevant to the evaluation techniques we have designed, our assumption is that it is in fact a key feature. Rather than just producing an improvement in a single variant of either DTED or DERP, we would thus expect it to produce dramatically improved results in all variants of both tools. The expectation of widespread improvement is encoded in hypothesis H^{STRUC}_{ALL}. In a very similar way, H^{ORD}_{ALL} broadens the assumption put forward in H^{ORD}.

7.1.3 Secondary investigations

In addition to our primary investigations, summarised by the hypotheses presented above, a number of features of our experiments may be of interest when considering broader aspects of language and evaluation than those codified thus far.

The first of these is how much human judgments are consistent across different judge demographics. While not integral to our questions of automatic evaluation of translations, an investigation into the effect of age, education level, languages spoken and other factors could further our understanding of the factors affecting human perception of both ordering specifically and adequacy in general.

Our main priority when responding to the above question is, however, ensuring that the data we have gathered is as consistent as possible *despite* such factors. As such, in Section 7.2.3 we discuss a method for controlling for participant variation. We then break down the information we have gathered by the three factors mentioned above. We briefly discuss the relevance of this breakdown to our own conclusions in Section 7.3.5, and report the data itself in Appendix B, pages 180 to 185.

The other investigation permitted by our data yet not directly integral to our primary hypotheses is related to automatic permutations. As described in Section 6.2.3, such sentences exist first to allow us to verify that our human judgments follow expected trends, and second to observe the performance of our tools when faced with more homogeneous, predictable error types than those existing in real translations.

As mentioned earlier, we combine these two investigations in Section 7.3.3 and discuss the implications of our observations in Section 7.4.2. Note that as automatic permutations are a somewhat unknown quantity in our experiments, we do not propose any formal hypotheses relating to them. We instead simply observe the consistency of performance of different tools when applied to them, with a view to providing a dataset and set of preliminary conclusions which may be of benefit to future researchers.

7.2 Evaluating WOJ-DB

Before we can claim WOJ-DB to be a worthwhile and relevant corpus of human judgments, and consequently use it to evaluate our automatic evaluation tools, we must assess whether it meets the standards we would require of any generic corpus of judgments.

These should fit with its goal of being flexible to a wide range of uses as per Section 6.2.1, allowing it to be incorporated to both our and others' future research projects.

It is important to note that we do not discuss the results or content of the pilot study (Section 6.4) in this chapter. Visual descriptions of that experiment can be found in Appendix A, pages 160 and 161.

We consider two aspects of WOJ-DB to be most relevant to this assessment. First, its internal characteristics of sentence quality, language, length and so on must be varied, relevant and evenly distributed. These aspects of the survey were discussed earlier, in Section 6.3.5.

Second, the human judgments we received should as much as reasonably possible respect the conflicting priorities of internal consistency and demographic variation, while containing judgments which are reliable yet remain representative of a non-expert reader audience. These elements are more related to our participants' responses than the survey's composition.

We use all participants' responses to investigate broad characteristics of WOJ-DB in Section 7.2.1, adjusting for individual variations as discussed in Section 7.2.3, while specific subsets of participants are discussed in Section 7.3.5. We also investigate the extent to which the scores we have gathered correlate with quality judgments from other sources, in Section 7.2.2.

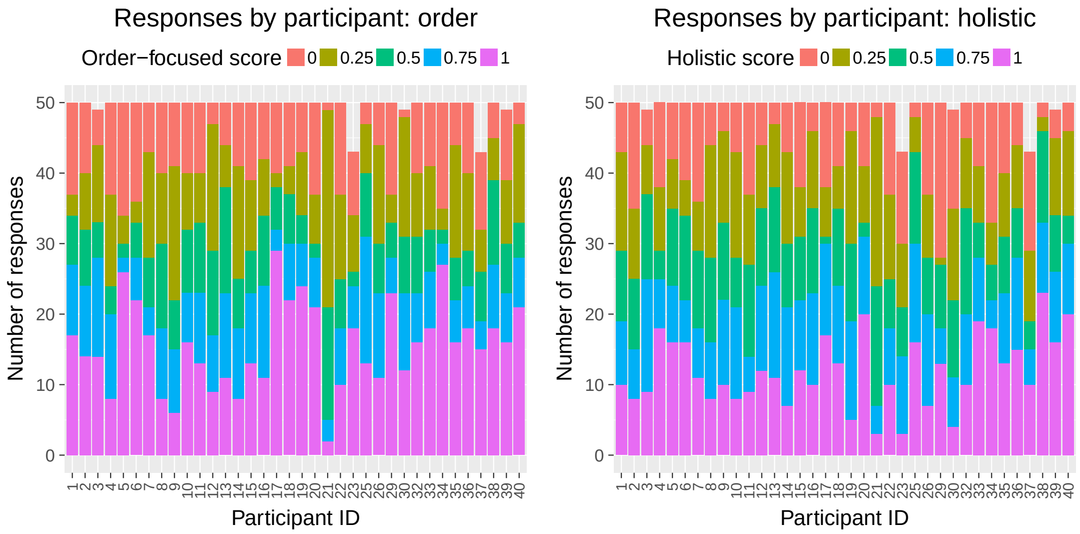


Figure 7.1: Distribution across participants of different score levels

7.2.1 Broad response trends

While the controllable aspects of the survey – sentence quality, length and other features – are investigated in Section 6.3.5, the most relevant feature of the survey is of course the participants' responses.

Figure 7.1 shows how participants varied in their responses to the two questions they were asked about each sentence. It contains two graphs indicating the distributions of

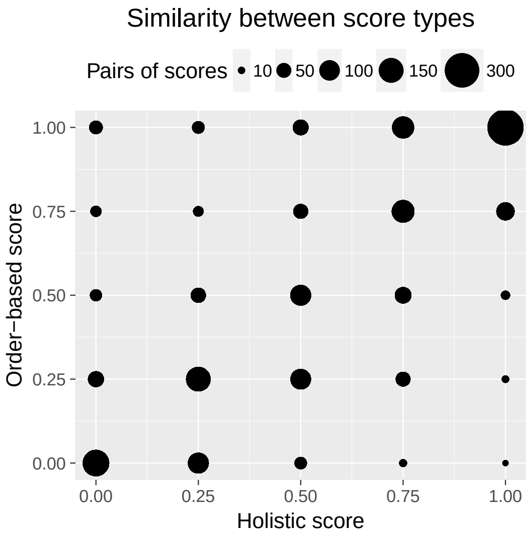


Figure 7.2: Similarity between holistic and order-focused adequacy scores in WOJ-DB

each of the five possible numericised scores for these questions. Note that not every participant responded to every question. Two participants simply omitted single questions, while in one case an entire page was skipped.

We can see in the graphs that all participants used every score available. Additionally, though some participants disproportionately used or avoided specific scores, in most cases the scores were relatively evenly distributed. The primary exception to this was that the word order of sentences was particularly often described as perfect, with a maximum score of 1.

We do not consider this deviation surprising because of the ways in which sentences were chosen. The random selection process for WMT-sourced hypothesis sentences, for example, allows for the inclusion of many sentences for which order simply happens not to have been a problem. As for permuted sentences, the 'choice' algorithm is guaranteed to retain the base translation's correct word ordering, suggesting that for around 25% of permuted sentences the order score should be perfect.

Using Figure 7.2 we investigate how the two types of participants' scores varied relative to each other. The X axis indicates the scores provided, by all participants and for all sentences, for the first, broad question, while the Y axis shows the same for the second, order-specific question. The size of the points indicates how many times the given scores were both given for the same sentence: thus, the largest circle in the top right-hand corner indicates that 323 sentences were assigned both a perfect holistic score *and* a perfect order-focused score, compared with 168 sentences which received the lowest scores for both features as represented by the bottom left-hand circle.

The primary indication given by this graph is that the two score types had similar trends. Indeed, the Pearson correlation coefficient between the two is a moderate-to-high 0.605 when considering hypothesis translations only, or 0.699 when applied to automatically permuted sentences. Thus, a given value for one score can be used as a somewhat reliable predictor of the value for the other score, though far from universally. Note that there were noticeably more sentences with very low overall score which were considered to be unaffected by incorrect word order (39 in total) than the mere 6 sentences where the word order was extremely muddled but the overall sentence was completely clear.

7.2.2 Correlations

While the scores produced by participants are important in their own right, they gain relevance when compared to scores produced by other techniques. In pursuit of our goal that WOJ-DB be verifiable (Section 6.2.2), through such comparisons we can gain a limited understanding first of whether the judgments it contains are reasonable: i.e. whether they conform to simplistic automatic evaluations. We can then compare the scores in WOJ-DB to those of our own tools, DTED and DERP, to evaluate the success of these last.

We begin by establishing a number of baseline methods for evaluating sentence quality. These fall into two categories: bespoke machine translation evaluation tools, and more generic statistical or human observations.

Recall that in Section 3.5 we described a number of such statistical techniques. We have calculated the percentage of aligned words, and Kendall's τ correlation coefficient between those alignments, for each pair of sentences in WOJ-DB.

In addition, we have calculated scores for two popular metrics: BLEU [Papineni *et al.*, 2002] and Meteor [Lavie and Agarwal, 2007]. See Section 2.2.1 for information on how these tools work. Given their open-source nature we have been able to run them on every sentence in WOJ-DB: both WMT hypotheses and automatic permutations. We have also run the 'chunking' component of Meteor (Section 3.5.4) in isolation and reported its correlations as 'Meteor (chnk)', allowing comparison between the scores of the off-the-shelf tool and its word-order-related sub-tool.

We have compared all of these scores, in addition to the normalised WMT ranks and permutation degrees, to the judgments provided by participants in WOJ-DB. For all such comparisons we have used Spearman rank correlation coefficient (Spearman's ρ) [Spearman, 1904], a common technique for measuring shared variation. It ranges from 1, indicating perfect agreement between the two score types, to -1 for a pair of score sets whose orders are exactly opposite.

Unlike the marginally more common Pearson's correlation coefficient [Koehn, 2010, p. 230], Spearman's ρ considers only scores' relative ranks and not their absolute values. This allows it to be unaffected by differences in the distributions of scores produced by different tools: should one tool be broadly linear but another cluster a high proportion of its scores towards a particularly low value, for example, the Pearson correlation between the two scores will be diminished relative to Spearman's ρ for the same dataset. As we are primarily interested simply in how useful a translation is relative to others, rather than in the exact distributions of scores produced by our highly heterogeneous scoring techniques, we consider such variation unhelpful.

If the reader is interested in the ranges of absolute scores produced by DTED and DERP, these can be found in Appendix B, pages 172 and 173.

Figure 7.3 shows the correlation coefficients between each tool and the two types of judgment collected in WOJ-DB, with comparisons against order-focused adequacy shown on the left and more holistic adequacy on the right. Third-party tools are shown in green, while all others are in purple. Note that both the percentage of aligned words

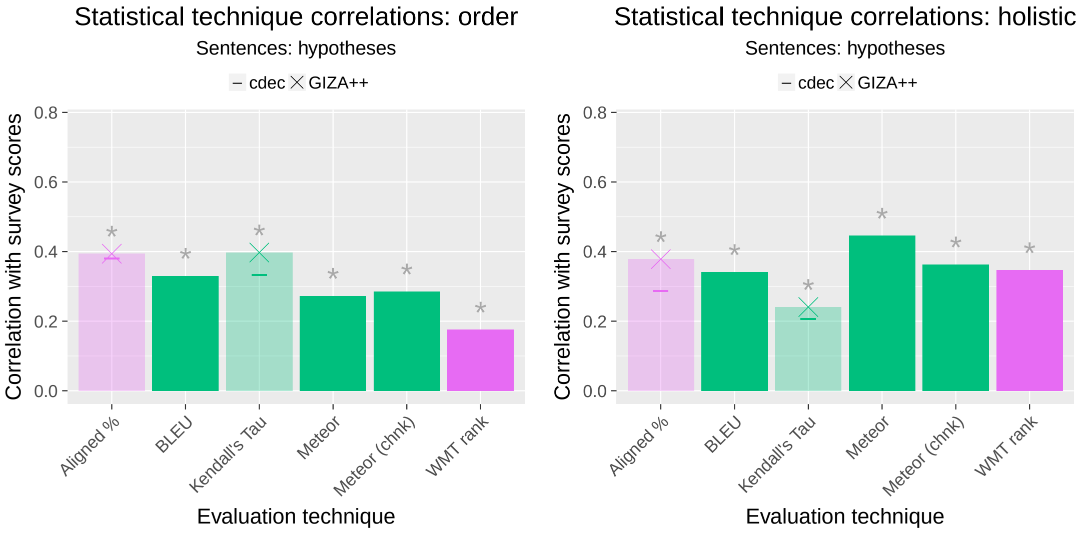


Figure 7.3: Correlations between scores in WOJ-DB and those calculated by automatic metrics and produced as part of WMT. Colours are summarised in Table 7.2 (page 117).

and our use of Kendall's τ are dependent on the tool used to align the sentences. We have produced separate scores using each of the two aligners used with DTED and DERP: these are reported separately in the graph, overlaying a bar whose height in all cases indicates the highest obtained correlation for the given technique.

To ensure that the conclusions we draw are reliable, we have also run statistical significance tests for the correlations. The results of these are indicated in the graph: for any correlations which are significant at the 0.05 level (here, all of them) an asterisk (*) is shown above the appropriate bar. If multiple configurations are used, this indicates significance for *all* correlations reported. Note that while correlations are based on scores adjusted according to participants' random effects, as described in Section 7.2.3, significance tests have been run on the unadjusted data for increased rigour.

We can see from the graphs that scores in WOJ-DB generally match those of our comparison tools with a medium level of correlation [Koehn, 2010, p. 230]. Off-the-shelf Meteor conforms to our expectations by correlating better with holistic judgments than those focused on word order, while its chunking component performs infinitesimally better than the main tool on order-focused scores instead.

While correlations in the vicinity of 0.3 are far from strong, they are within a range which is reasonable given our domain [Lavie and Denkowski, 2009; Fishel *et al.*, 2012b; Stanojević and Sima'an, 2014a]. Correlations of this order are a necessary consequence of our decision to focus on sentence-level scores rather than system-level ones, as discussed in Section 3.3.3.

While the statistical strength of the conclusion is rather limited, based on these correlations we consider that WOJ-DB has broadly succeeded at its goal of containing reasonable judgments on two fundamental aspects of machine translation.

7.2.3 Participant variation

In pursuit of our goal of verifiability (Section 6.2.2), we have designed each questionnaire in WOJ-DB such that a portion of the sentences included occur also in every other survey (see Section 6.3.4 for more information). In this manner, we can compare the results of different surveys to account for the variation inherently associated with human participants.

This variation is caused by the simple fact that not all people approach the questions of sentence quality in the same way. Coupled with this, not all may understand our five-point system for scoring a sentence. Some may score most sentences highly except for a few with particularly low quality, while others may do the opposite.

To evaluate these trends, we have produced a linear regression relation [Galton, 1886] between each score type and the two sets of scores provided for all participants. We have then extracted the random effects [Laird and Ware, 1982] from these relations, representing for each tool how much all scores were affected by other factors. Of these, we selected only the random effects associated with the participant's unique identifier, and adjusted all scores by the appropriate value from this list for that participant.

In this manner, we have attempted to control for the variation between participants which exists in our dataset. To evaluate whether this procedure made any significant changes to the scores, we have run a two-tailed Student's t-test [Student, 1908] between the original and adjusted scores. With a p-value of 0.118, the differences were not significant at the 95% level. This indicates simply that while the scores were affected by the adjustment, the changes were limited and the adjusted scores are still comparable with the originals.

To further investigate the success of our technique, we have plotted the correlations with baseline statistics for both the adjusted dataset (Figure 7.3) and the original scores (Figure 7.4).

While the difference in correlations observed is minimal, it is nonetheless consistent and positive. All the baseline tools discussed in Section 7.2.2 observed either small improvements or no change between the two techniques. As a result of this, we consider that the adjusted scores represent true word order marginally better than the original scores, and have used them when calculating all other correlations discussed in this chapter. Correlations based on the original, unadjusted scores can be found in Appendix B, pages 174 to 177.

7.3 Evaluation of metrics using WOJ-DB

While it is important to investigate whether the scores contained in WOJ-DB are consistent and able to correlate to a reasonable extent with simple statistical tools, this is not the primary use for the database. Although we consider it important for its data to be applicable to other research projects, it is principally for the evaluation of structure in machine translation that it was created.

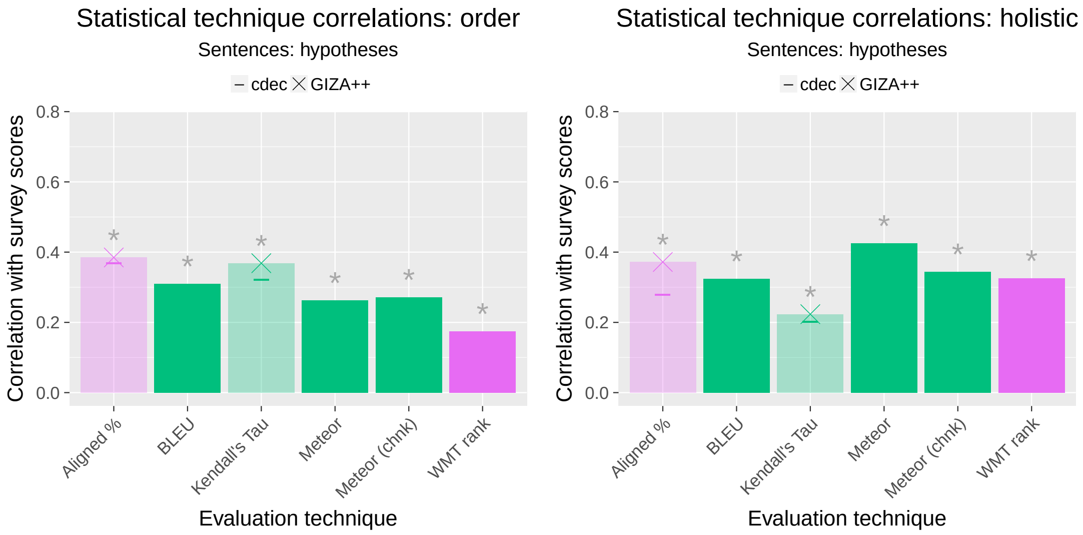


Figure 7.4: Correlations between scores in WOJ-DB and those calculated by automatic metrics and produced as part of WMT, without adjustment for random effects. Colours are summarised in Table 7.2 (page 117).

7.3.1 Third-party tools

Before inspecting the correlations between our own tools and the judgments contained in WOJ-DB, we have one further set of baselines to consider. We consider the scores submitted by various tools to all relevant WMT evaluation tasks: that is, those from 2014 to 2016 inclusive. We have produced correlations between the scores submitted to WMT and those produced for WOJ-DB relating to word ordering, for the varying numbers of sentences included for each tool in both evaluation sets. We omitted scores for any tools for which fewer than 15 scores existed in both WMT and WOJ-DB.

These correlations are shown in Figure 7.5. Tools whose scores were retrieved directly from WMT are indicated in blue, while tools which we have run on every sentence in WOJ-DB are considered in green as before, for comparison. Table 7.1 repeats this information while also indicating how many scores for each tool related to sentences included in both WOJ-DB and the relevant year(s) of WMT.

Most of these tools achieve virtually negligible correlations. However, due to the lack of statistical significance for most tools we believe this to be a result of the limited data available. With relatively small datasets for each tool, the impact of random variations may be important to the point where data are unreliable. This supposition is supported by the fact that all six tools whose correlations are statistically significant at the 95% level, and from which vastly more scores were available, perform dramatically better than almost all others.

It should also be highlighted that tools submitted to WMT were not designed for the exact situation in which they are here being evaluated. Specifically, while we are investigating the quality of word ordering in terms of adequacy, most tools were simply intended to measure relative general quality, as interpreted for example by judges at WMT.

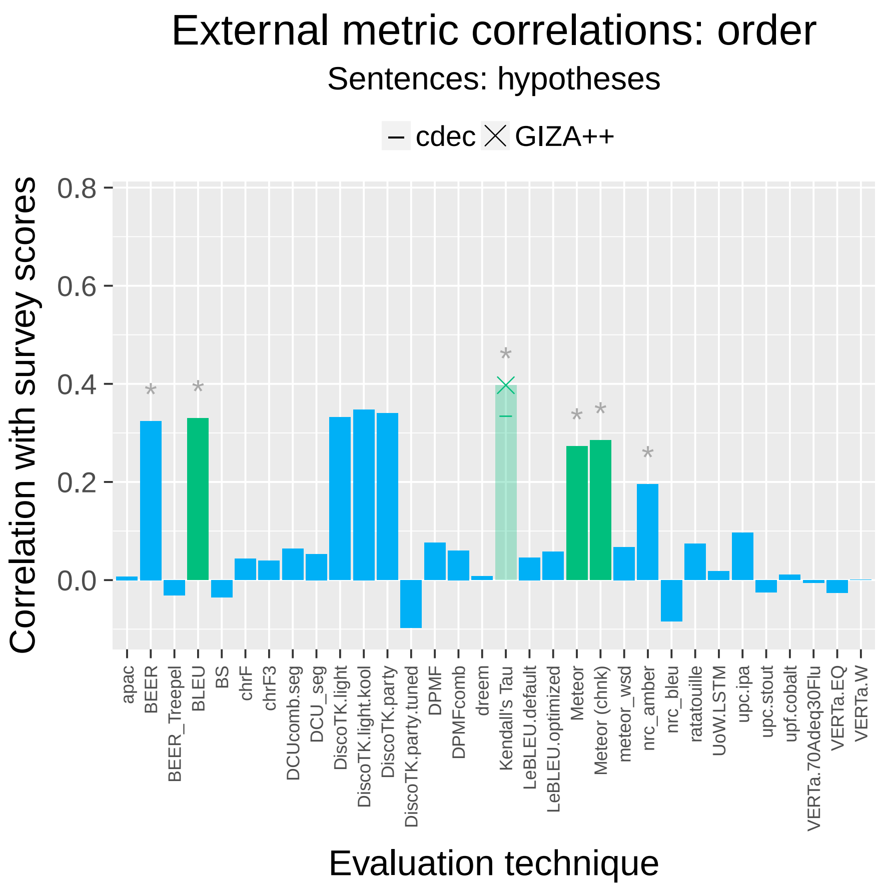


Figure 7.5: Correlations between WOJ-DB human judgments on word order adequacy and external scoring systems submitted to WMT. Colours are summarised in Table 7.2 (page 117).

There are three exceptional tools among those submitted to WMT which perform to levels comparable to those of BLEU and Meteor. These are BEER [Stanojević and Sima'an, 2014a], AMBER [Chen and Kuhn, 2011] and DiscoTK [Joty *et al.*, 2014]. Of these, BEER has the most datapoints of any of the tools we are discussing, covering 63% of sentences in WOJ-DB, while the best-performing variants of DiscoTK have the fewest at just 5%.

It is interesting to note the range of approaches between these three tools. Specifically, AMBER represents an updated yet traditional approach, focusing on precision with a length-related penalty similar to BLEU. By contrast, DiscoTK represents a foray into the use of discourse structure and tree kernel comparison, a much more recent technique. BEER combines the two, with components based on structure through permutation trees, alongside more traditional n-gram comparisons.

We consider that all three of these results represent encouraging trends for the tools in question, albeit trends which cannot be fully understood without more scores being available for both them and their competitors. We leave a fuller investigation of third-party tools within WOJ-DB as future work.

Tool	Count	%	Corr.	p-value
apac	138	25.3	0.007	0.992
BEER	342	62.6	0.324	0.000*
BEER_Treepel	204	37.4	-0.025	0.721
BS	204	37.4	-0.012	0.870
chrF	204	37.4	0.043	0.900
chrF3	204	37.4	0.039	0.811
DCUcomb.seg	138	25.3	0.066	0.274
DCU_seg	138	25.3	0.055	0.365
DiscoTK.light	26	4.8	0.332	0.131
DiscoTK.light.kool	26	4.8	0.348	0.334
DiscoTK.party	26	4.8	0.340	0.386
DiscoTK.party.tuned	49	9.0	-0.098	0.504
DPMF	204	37.4	0.076	0.368
DPMFcomb	204	37.4	0.060	0.530
dreem	204	37.4	0.008	0.917
LeBLEU.default	204	37.4	0.046	0.579
LeBLEU.optimized	204	37.4	0.058	0.571
meteor_wsd	204	37.4	0.068	0.500
nrc_amber	138	25.3	0.196	0.001*
nrc_bleu	138	25.3	-0.084	0.167
ratatouille	204	37.4	0.074	0.503
UoW.LSTM	204	37.4	0.018	0.837
upc.ipa	138	25.3	0.096	0.117
upc.stout	138	25.3	-0.025	0.627
upf.cobalt	204	37.4	0.011	0.777
VERTa.70Adeq30Flu	204	37.4	-0.006	0.672
VERTa.EQ	342	62.6	-0.026	0.561
VERTa.W	342	62.6	0.001	0.994

Table 7.1: The number of scores for each metric extracted directly from WMT [Bojar *et al.*, 2014, 2015, 2016a], along with the proportion (%) of unique hypothesis sentences in WOJ-DB which were scored by each metric. *Corr.* and *p-value* refer to the correlations reported in Figure 7.5; * indicates significance at the 95% confidence level.

Colour	Description
Red	Variants of DERP
Yellow	Variants of DTED
Green	Third-party metrics with scores for all sentences
Blue	Third-party metrics with scores for some sentences
Purple	Other evaluation methods
Grey (*)	All correlations significant at 95% level

Table 7.2: Meanings of the various colours used within correlation graphs in this chapter

7.3.2 DTED & DERP

Having established the baselines both of statistical techniques such as Kendall's τ and of various tools submitted to WMT, we can now meaningfully inspect the scores of our own tools. To that end, we have produced Figure 7.6, showing correlations with WOJ-DB order-specific scores for each experiment performed using both DTED and DERP in addition to the off-the-shelf tools we include for comparison.

Experiments are split by three features: flag configuration (Sections 4.3 and 5.10), third-party tool setup (Sections 3.4, 4.2.1 and 5.9) and type of sentence being investigated (WMT score or permutation). Each of these is indicated in the relevant graphs, while a summary can be found in Table 7.3. Note that flattened versions of our tools do not rely on either a tagger or the parser, 'b' DTED ignores information from alignments and 'fc' makes no use of the information provided by any of these three tools.

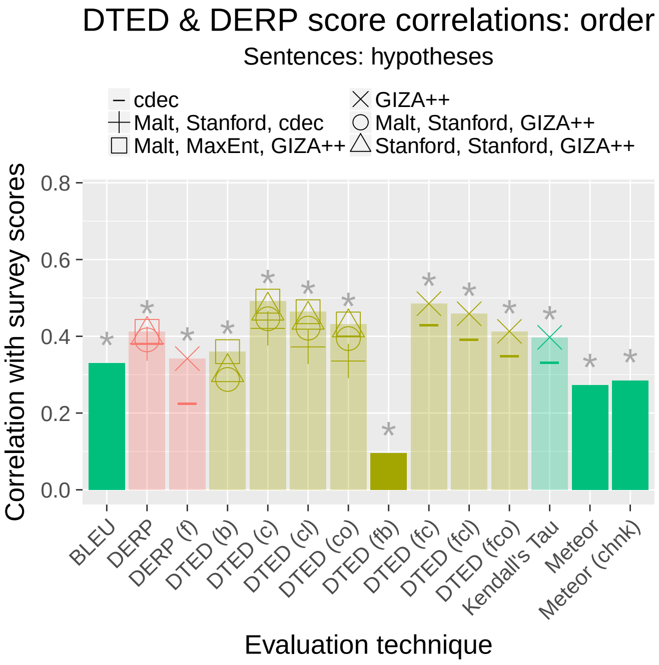


Figure 7.6: Correlations between WOJ-DB human judgments on word order adequacy and variants of DTED and DERP. Colours are summarised in Table 7.2 (page 117).

While all meaningful combinations of DTED flags and both variants of DERP have been evaluated separately, we have limited the number of combinations of third-party tools. This is primarily because those tools are not the main subject of our investigation: we simply wish to ensure that none of the three utilities necessary for the tools' functioning significantly affect their scores. We have thus produced variations on a single 'default' configuration – the Malt parser using the Stanford parser's internal tags and alignments from GIZA++ – and ensured that exactly one alternative configuration exists as a comparison group for each component.

We can observe a number of trends from the information in these graphs, each discussed in more detail in Section 7.4. First, if arguably least important: we can confirm that the choice of third-party tools does not dramatically affect the efficacy of either

Configurations		
DTED & DERP		
Parser	Tagger	Aligner
Malt	Stanford	GIZA++
Malt	NLTK MaxEnt	GIZA++
Malt	Stanford	cdec
Stanford	Stanford	GIZA++

×

Variants			
DTED		DERP	
Flags	Section	Flags	Section
b	4.3.1	*none*	5.6.1
c	4.3.1	f	5.10
co	4.3.2		
cl	4.3.2		
fb	4.4.1		
fc	4.4.1		
fco	4.4.1		
fcl	4.4.1		

Table 7.3: All executed configurations and variants of both DTED and DERP. Note that flattened versions (with flag ‘f’) do not make use of either parser or tagger.

DTED or DERP. While in the case of flattened DERP the alignment system results in a variation in correlation of 0.114, for structured versions of both tools the variation due to third-party configuration is no higher than 0.105.

The most encouraging trend we can see in the data for both tools is that they perform, for the most part, noticeably better than the off-the-shelf baselines they are compared to. With just a single exception in the case of ‘b’ (and fb’) DTED, almost all variants of both tools achieve a correlation higher than BLEU, the highest-scoring off-the-shelf tool run on the entire dataset, ranging from 0.004 below BLEU to 0.161 (48.8%) above it. When considering all other scores included in WOJ-DB, the highest-performing was in fact Kendall’s τ, though the highest-scoring of our tools (‘c’ DTED) still exceeded this by up to 0.094 (23.7%).

With the wider range of flags in DTED than DERP, we can observe some interesting variations when different sets of flags are applied to the former. The most noticeable of these is the dramatically poor performance of ‘b’ and fb’ DTED relative to the others. We do not find this surprising, as these are the variants with the least information available to them: as discussed in Sections 4.3.1 and 4.4.1, the former provides only a very simplistic measure of similarity between trees, while the latter is unable to measure anything beyond sentence length.

More intriguing are the lower correlations associated with ‘o’ and ‘l’ DTED, and their flattened counterparts, relative to the simpler ‘c’ DTED. This trend is clear across all configurations of third-party tools, and even between flattened and structured versions of DTED. It strongly suggests that the behaviour introduced in these versions – that is, a strict prioritisation in the matching of structures relating only to aligned nodes – is counterproductive to the attempt to predict human judgments.

The most notably surprising message contained within DTED’s correlations is to do with the performance of flattened versions relative to their structured cousins. While the flattened tools do not strictly fare better than the structured ones, in almost all cases they perform to a closely comparable level. The only exception to this is ‘b’ (and ‘fb’) DTED, which as discussed earlier is the least interesting variant for our purposes.

The trends shown by variants of DTED are noticeably different from the performance of DERP. Notably unlike DTED, the correlations for the flattened version of DERP are noticeably lower than those for the structured default version. While the reduction is only slight in the case of the higher-correlation flattened version using GIZA++, even in that case it is 0.0097 (2.7%) lower than the lowest-correlating configuration of structured DERP.

The other noticeable difference between correlations for DTED and for DERP is that contrary to our expectation, DERP is unable to outperform any variant of DTED with more complexity than the basic ‘b’ flag. While structured DERP still correlates more highly than BLEU, the performance of its highest-correlating configuration is still 0.079 (16%) lower than the highest correlation relating to DTED.

7.3.3 Automatic permutations

One more important aspect of WOJ-DB, which both requires its own evaluation and permits insights into the working of other tools, is its automatic permutations. Thus far we have omitted such sentences, considering only machine translated hypothesis sentences when producing the correlations we have presented.

Correlations related to automatic permutations are shown in Figure 7.7. The same colour scheme, summarised in Table 7.2, is used to differentiate the various types of tools, of which we include all those available.

We see a number of differences between these graphs and those related to WMT hypothesis. We will discuss the observable trends and their possible reasons in this section, before examining their more general implications in Section 7.4.2.

Prime among these observations is that the variation in correlation between different tools is somewhat diminished: third-party tools such as BLEU and Kendall’s τ perform for the most part on a par with our own, or even exceed the correlations of DTED and DERP.

‘order’, ‘swap’ and ‘choice’ permutations

In the case of ‘order’ and ‘swap’ permutations, the trends within our own tools are largely unchanged. In keeping with judgments for hypothesis translations, the flattened version of DERP is noticeably less reliable in predicting human judgments than its structured counterpart. The flattened versions of DTED can, on the other hand, be seen to compare closely with the structured versions, with the exception of ‘b’ and ‘fb’ DTED.

In the case of ‘order’ permutations, ‘fb’ DTED has little information available to it with which to make a judgment. As no words are inserted or deleted, yet as described in Section 4.4.1 all score variations arise from disparities in sentence length, the only variation between sentences when excluding alignments and dependency structure is caused by rare and (usually) unimportant alterations of punctuation performed by the generation algorithms. In most cases, when words are moved any punctuation associated with them is also moved, but it is not rare for certain characters to be overwritten, resulting in a different number of tokens being parsed and leading to the length disparity detectable by ‘fb’ DTED. Such differences occur in 58% of permuted sentences.

In the case of ‘swap’ permutations, variations in sentence length are due to even rarer edge cases than those of ‘order’ permutations. No punctuation is overwritten while

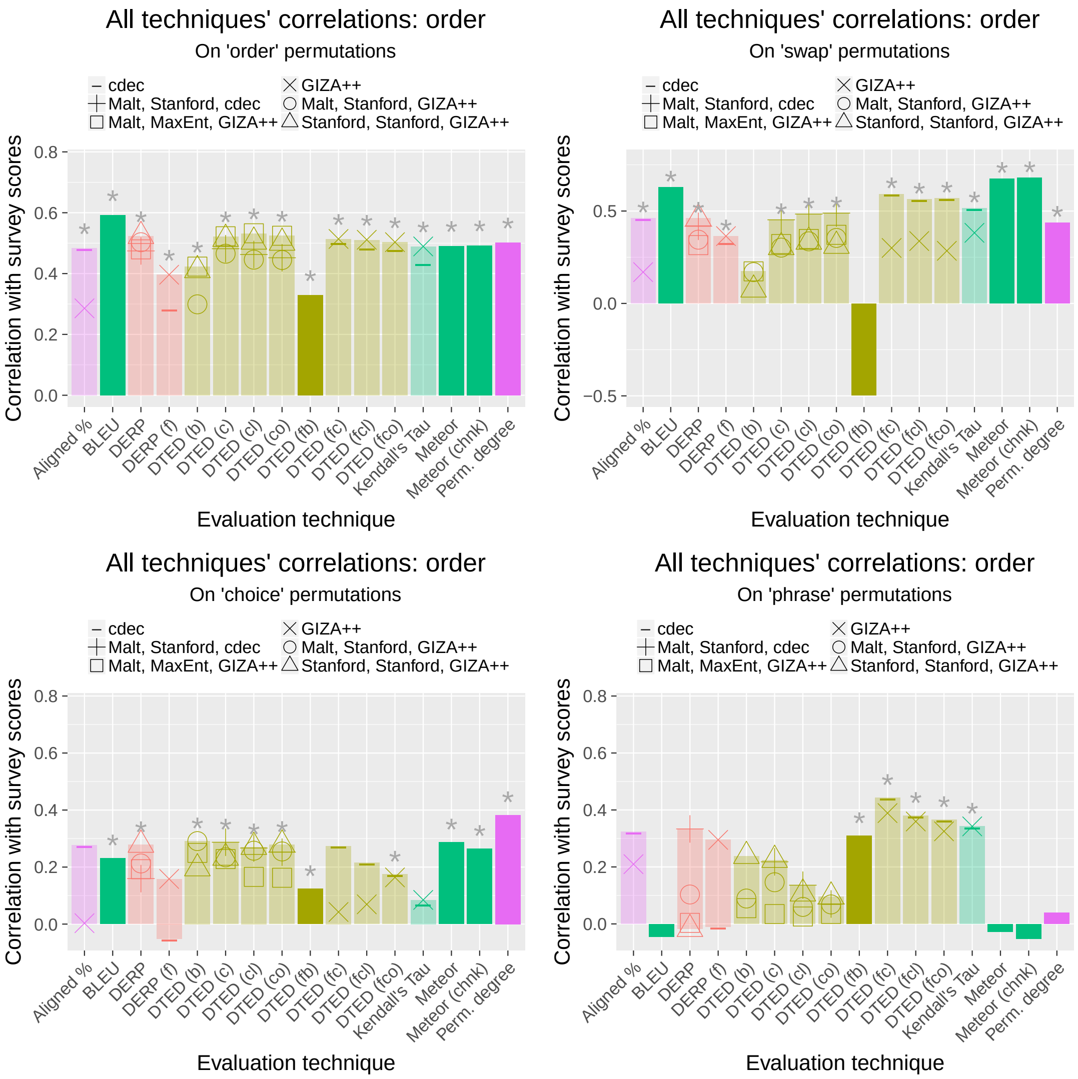


Figure 7.7: Correlations between WOJ-DB human judgments on word order adequacy and other techniques across automatically permuted sentences in WOJ-DB. Colours are summarised in Table 7.2 (page 117).

generating these, resulting in just three datapoints contributing to the negative – though not significant – correlation we see. These are again the result of special interactions between punctuation and tokenisers, e.g. in one sentence where “No.” and “lot” in a reference are swapped to “lot.” and “No”: in the former case, “No.” is treated as one token due to rules of English grammar, while in the latter case “lot.” is separated into two separate tokens. These three datapoints result in a negative trend in the data.

Similar edge cases result in the correlations shown for ‘phrase’ and ‘choice’ permutations, for which respectively 13% and 6.8% of permuted sentences varied from the usual score of 0.5 for this variant of DTED.

In the case of ‘choice’ permutations, we see similar trends to ‘order’ and ‘swap’ permutations with one main exception: almost all correlations are simply reduced, to levels below even those calculated on hypothesis translations. For our own tools this is not surprising, as the intention behind them was exclusively to measure word ordering rather than choice of words.

However, for other tools – both third-party machine translation evaluators and other statistical techniques – the dip in performance is more surprising. We believe that this is a result simply of the reliance of such tools on simplifications from exact word choice.

For example, Meteor relies on stemming to match words together, thus ignoring the occasional word replaced during ‘choice’ permutation which retains the same stem as an original word. This may not accurately reflect the confusion for humans which the change may cause. Similarly, Kendall’s τ relies exclusively on alignment, which in turn may in many cases use only a word’s tag, rather than a specific word form, to determine its aligned partner(s).

‘phrase’ permutations

The final permutation type, ‘phrase’, results in more deviation than the other three from the trends we have thus far discussed. To begin with, it appears that the prediction power of the three off-the-shelf machine translation tools was entirely negligible. While the correlation is not significant at the 95% level, we do not in any case consider this entirely surprising.

Both BLEU’s focus on n-grams, and Meteor’s examination on adjacent chunks of words, can be ‘gamed’ by such permutations. The movement of an entire phrase at once results in the words belonging to that phrase retaining their relative positions to each other: thus, the only n-grams or chunks which are affected by a phrase movement are those at the boundary between the phrase and its larger context. The small number of n-grams which are disrupted by such phrase movements is thus unlikely to give rise to a real measure of how severe the movement of a phrase is, yet given the exact matches between words in each pair of sentence it is virtually the only feature which can vary the scores produced by either BLEU or Meteor.

We note that the ‘degree’ of ‘phrase’ permutations correlates to a similarly negligible extent with human judgments. In this case, we believe the reason to be simple: the ‘degree’ refers specifically to the number of words moved, while in practice the quality of the sentence relies on entirely other factors. Specifically, the impact on order quality of a phrase’s movement is related mostly to the choice of phrase moved and the destination of that phrase. Neither of these factors are controlled for during generation, and thus neither are captured in the ‘degree’.

As for our own tools, we can see that in all cases involving structure, the choice of tool used to generate that structure dramatically affected the ability of both DTED and DERP to predict sentence quality of ‘phrase’ permutations. We hypothesise that as the choices of moved phrases and target locations are made randomly, sentences permuted through this technique can have ambiguous or confusing structures. Given that such complexities do not necessarily have anything to do with the real-world complexities of English grammar, there is no guarantee that any parser or tagger will be able to extract a meaningful interpretation from the sentences.

Given this, different such tools will be more or less able to produce parses which happen to allow DTED and DERP to produce a score which a human would consider reasonable. Effectively, the unusual nature of the sentences results in a great deal of noise in the parsing process. This is reflected in the lack of statistical significance for variants involving structure. Having said that, when trees are flattened such noise is eliminated, resulting in a more straightforward task for our tools and hence a better correlation with human judgments.

7.3.4 Holistic scores

We have thus far discussed the implications of the correlations between various score types and the human judgments we have gathered on word order quality. However, in addition to such judgments, during the survey we also asked a second question. Discussed in Section 6.2.1 and intended to aid our response to the key question presented in Section 3.2.3, this second question related to the overall adequacy of the sentences in question.

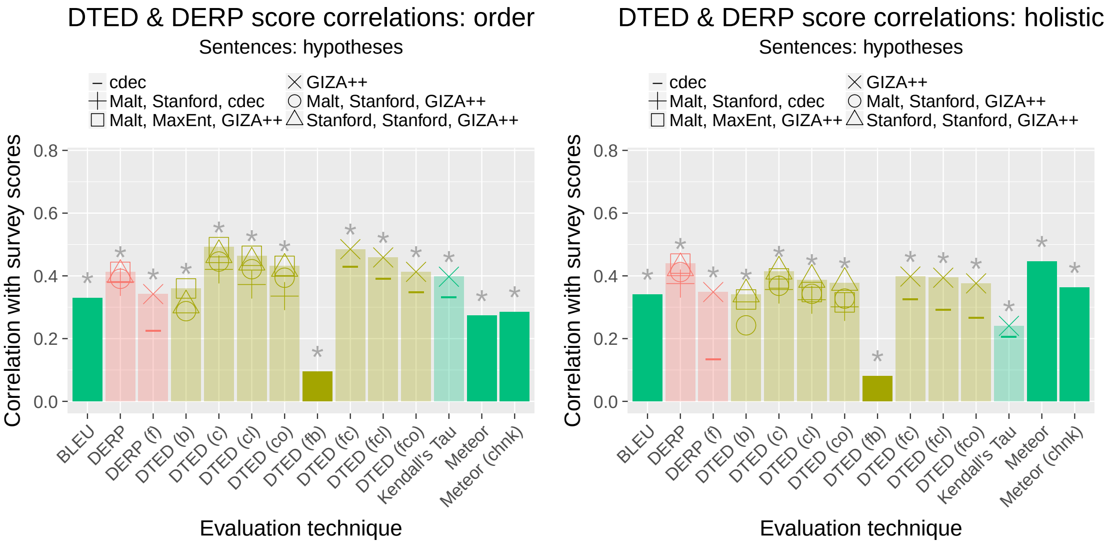


Figure 7.8: Correlations between WOJ-DB human judgments on both order-focused and holistic adequacy and variants of DTED and DERP. Colours are summarised in Table 7.2 (page 117).

Figure 7.8 shows the correlations between responses to this more holistic question and all variants and configurations of DTED and DERP, while also repeating the information shown in Figure 7.6. We observe two primary differences between the two graphs: the comparison tools shown in green perform to a marginally higher level when compared to holistic judgments; and ours perform slightly less well.

The reduction in the efficacy of our tools is not absolute, in that for only two variants ('fb' and 'fc' DTED) the highest correlations with holistic human judgments are lower than the lowest correlations with order-focused scores: 0.081 to 0.095 and 0.398 to 0.407 respectively. Both of these variants use flattened trees. We discuss the implications of this in Section 7.4.

While it may be interesting to investigate the relative success at predicting adequacy of other tools than DTED and DERP with and without a focus on word ordering, we consider that only those indicated in Figure 7.8 are truly relevant to our experiments. This is because limited data is available for tools not included, weakening any conclusions we could draw. Comparative performance of all available tools are presented in Appendix B, pages 178-179.

7.3.5 Participant variation

While the correlations shown and discussed thus far include scores from all participants, it is possible that not all such data is relevant. Before using WOJ-DB to evaluate other tools, such as DTED and DERP, we must briefly discuss the variability in scores between participants.

Recall from Section 6.5.2 that we recruited participants with a range of ages, genders, education levels and linguistic backgrounds. In addition, a single participant was dyslexic. While the effects of these factors may be controlled for by per-participant random variables as discussed in Section 7.2.3, it may be interesting to inspect in detail the differences between demographic groups.

To this end, we have produced a number of reports similar to Figures 7.3 and 7.7, showing the correlations with human judgments of most tools when considering only scores produced by participants belonging to specific demographics.

While these graphs contain interesting information about different aspects of our participants, they are not considered key to our investigation. This is for two reasons. First, the act of extracting subsets of datapoints from those available to us must necessarily reduce the reliability of our conclusions. With for example just 25% of our participants having ages over 30, any conclusions drawn on that or any subset will be more subject to random chance and thus weaker than those based on the dataset as a whole.

The second reason why we do not place a high importance on correlations based on participant subsets is to do with our goals: quite simply our experiment was not intended to investigate these variations in detail. Results based on them can thus provide at most peripheral information for our conclusions.

Note that while the differences between graphs may not be integral to our own investigations, it is important to observe whether they nonetheless indicate the same trends we have observed on the entire participant pool. While some variation is inevitable, especially with demographics to which only a small number of participants belong, we would expect the trends we have observed throughout Section 7.3 to broadly hold true for all subgroups.

Happily, we believe that such trends can indeed be observed across all participant subsets we have investigated. In all cases DERP was able to better predict human judgments when provided with dependency structures, while the same was conspicuously not the case with DTED. In most cases, the same tools whose scores have been extracted from WMT perform the highest. In addition, the approximate strength of the strongest correlations remains broadly constant.

Given the general agreement in trends between the correlations presented thus far and those relating to different participant subsets, and the statistical limitations of any investigation using subsets of the data available to us, we do not present the graphs in this chapter. Should the reader be interested, they are directed to Appendix B, pages 180 to 185.

7.4 Discussion

Having investigated the appropriateness of WOJ-DB, the relevance of our tools' scores to both it and third-party evaluation techniques, and the effects of constraining sentence variations or participant demographics, we must relate these separate threads to our key hypotheses and thesis questions.

We first discuss the scientific hypotheses from Section 7.1.1 in Section 7.4.1, before considering the broader answers we can draw from these to the questions outlined in Section 3.2.

7.4.1 Responses to hypotheses

For each hypothesis presented in Section 7.1, Table 7.4 indicates whether the null hypothesis has or has not been rejected for the experimental data we have collected.

We can see that of the first three major hypotheses presented in Section 7.1.1, H_0 has been rejected in each case. This is encouraging, suggesting that the overall thrust of our work was successful.

According to H^{EVAL}, we have succeeded in producing a tool which performs better when predicting human judgments of word order than any of the others we have used for comparison. This is a notable achievement, although it has limitations. First, between both our tools only one variant, 'c' DTED, was able to surpass the simpler Kendall's τ baseline across every single configuration of third-party tools.

An additional caveat to the success reported by H^{EVAL} is our limited number of tools used for comparisons. As suggested in Section 7.3.1, the sparse data available for tools submitted to WMT may have impacted the correlations reported for those tools. As such, only three existing tools were used for comparison: BLEU, Meteor and the chunking component of Meteor. To truly consider our work an improvement on the state of the art, we would need to compare it with more recent and varied third-party tools.

Despite these limitations, we consider the success of 'c' DTED an important marker of progress in the evaluation of word order in machine translation. Given how well-respected both BLEU and Meteor are, matching human judgments reliably better than

Hyp.	Result	Description of result
H^{EVAL}	H_0 **rejected**	Some variant of DTED & DERP ('c' DTED) outperforms all third-party/statistical techniques in WOJ-DB
H^{STRUC}	H_0 **rejected**	Some variants of structured DTED & DERP (DERP and 'b' DTED) outperform equivalent flattened variant
H^{ORD}	H_0 **rejected**	Order correlations overlap with or are better than holistic judgments for some (every) variant of DTED & DERP
H^{EVAL}_{DTED}	H_0 **rejected**	Some ('c') DTED outperforms all third-party/statistical techniques in WOJ-DB
H^{EVAL}_{DERP}	H_0 **not rejected**	Neither DERP outperforms all third-party/statistical techniques in WOJ-DB
H^{EVAL}_{DD}	H_0 **not rejected**	Neither DERP outperforms all DTED – in fact, all configurations of 'c' DTED outperform DERP
H^{EVAL}_{CONF}	H_0 **rejected**	Configuration can affect correlations by up to 0.114 (flattened) or 0.105 (structured), i.e. no more than 0.122
H^{STRUC}_{ALL}	H_0 **not rejected**	Not all structured DTED, DERP outperform flattened equivalents: 2/5 variants' ranges do not overlap
H^{ORD}_{ALL}	H_0 **rejected**	All DTED, DERP predict order quality at least as well as holistic adequacy

Table 7.4: Summary of hypotheses from Sections 7.1.1 and 7.1.2.

them across sentences judged by individuals from very varying demographics is a non-trivial achievement. Should others in the machine translation community agree, future work should be for us to package our tools in an easily accessible manner for publication.

Our second primary hypothesis pair, H^{STRUC}, relates to the merit of the inclusion of structure through dependency parses rather than relying on flattened 'trees'. While we do observe the effect we expected – a marked increase in the performance of both DTED and 'b' DTED when structure is included – it is the cases where this is not the case which are arguably more interesting.

As observed for H^{STRUC}_{ALL}, the performance of 'c', 'o' and 'l' DTED is comparable whether or not the tool has structural information available: that is, the highest-correlating configuration of the flattened variant performs better than the lowest-correlating configuration of the structured variant. We consider two alternative interpretations for this observation: first, structural information does not help these evaluation techniques (as queried in Section 3.2.2); and second, the noise introduced by flawed parses approximately outweighs any benefit provided by that structure. We discuss these alternatives further in Section 7.4.3.

The third and final of our primary hypotheses, H^{ORD}, is to do with the relative success of our tools at predicting order-focused and holistic quality. Our results provide a clearer response to this hypothesis than to the previous two: our tools perform equally well at predicting holistic judgments as those related specifically to word order. Only 'fb' and 'fc' DTED differ noticeably in performance when predicting the two quality types, with both of those variants performing better at holistic judgments. This is reflected in the rejection of our secondary, stronger $H^{ORD}_{ALL_0}$.

We now discuss the specific, secondary hypotheses presented in Section 7.1.2. Two of these, H^{EVAL}_{DTED} and H^{EVAL}_{DERP}, simply relate separately to the success of each of our two tools. As mentioned above, only one variant of our tools – 'c' DTED, manages to perform better across all configurations of third-party tools – although at least one configuration results in higher scores than any third-party system for seven of the ten variants of both DTED and DERP. While not as clear-cut as the success of 'c' DTED, this is nonetheless a positive result.

The non-rejection of H_{DD_0} represents a significant deviation from our expectations. As DERP was designed to take the strengths of DTED and build upon them (Sections 4.7, 5.1 and 7.1.2), the fact that DTED has actively performed to a higher level than its more complex cousin challenges our assumption that the detail it provides, in the form of inspection of relative paths, is helpful in evaluating ordering quality.

We believe that the reason for this is related to the surprising success of flattened versions of DTED. Discussed in more detail in Section 7.4.3, we posit that noise introduced by parsing low-quality sentences results in dramatically inaccurate dependency labels. While DTED ignores such labels and is able to make use of the broad structure of a tree, DERP's use of dependency labels forces it to overestimate all label-related errors to unpredictable extents.

This possibility is unfortunately not verifiable using the data available to us, which includes no way of measuring the noise or inaccuracy introduced by different factors. However, such information could be gathered in future studies, either by controlling or eliminating parsing noise (see Section 7.4.3), or by modifying DERP to minimise its reliance on dependency labels. For example, it could theoretically be altered to ignore all dependency labels and thus use only path lengths and relative directions in its calculations, bringing its functionality closer to that of Kendall's τ.

Finally, H^{EVAL}_{CONF} represents an investigation of the effect of configuration on our tools' performances. We hypothesised that the variation of parser, tagger and aligner would impact the effectiveness of neither DTED nor DERP by more than the standard deviation of all calculated correlations related to structured versions of those tools, and in practice saw variations within this bound. From this we can state that the choice of third-party tools is not a critical component of our tools.

Further than the requirements of our hypothesis, we can observe that even the highest variation between different tools, 0.114 in the case of flattened DERP, is just 27.6% of the highest correlation for that tool. Additionally, the variation caused by configuration was below the 0.1 threshold for seven of the ten possible variants of DTED and DERP. Contrasted with the disparity of 0.396 between the highest- and lowest-performing variants of DTED, we conclude that the effect of configuration on DTED and DERP is truly limited.

7.4.2 Automatic permutations

In addition to the hypotheses put forward in Sections 7.1.1 and 7.1.2, we have conducted a secondary investigation into the performance of automatic permutations. As discussed in

Section 7.3.3 we observe two trends from this investigation: heightened scores for almost all tools relative to machine-translated hypotheses; and lower disparities in performance between our tools and various third-party and statistical baselines than when considering machine-translated hypotheses.

We can ascribe both of these trends to the simplistic nature of our permutations. They were designed with the intention of representing consistent and predictable errors within sentences (Section 6.2.3). This very simplicity, a significant departure from the norms of natural language processing, allows us to draw more specific conclusions about the sentences than would be possible on more varied yet realistic translations.

Such simplicity and consistency, while diverging somewhat from the reality of machine translation, makes the problem of evaluation much more straightforward. A universally recognised difficulty in NLP is the sheer range of phenomena encountered in practice; this renders the task of capturing all possibilities using a simple computational algorithm close to impossible. However, when errors to be detected are themselves introduced by such an algorithm, it is to be expected that automatic tools perform to a (relatively) high level.

It is interesting to observe that while this trend seems strong in the case of two permutations, ‘order’ and ‘swap’, it is much weaker when we consider ‘choice’ and ‘phrase’ permutations. We believe the relative lack of success of evaluation of the latter type to be a result of its generation algorithm being more complex than the others’. The movement of entire phrases, without consideration for the meaning of such movement, treads an unfortunate middle ground between the simplicity of ‘order’ and ‘swap’ permutations and the complex yet meaningful variability of real sentences.

As for ‘choice’ permutations, the relatively low correlations reported for all tools have less of a clear interpretation. Most fundamentally, a lower performance is expected for tools designed to evaluate word order, as ‘choice’ permutations were not intended to contain any such errors.

This leads us to the primary goals of automatic permutations (Section 6.2.3). First, the heightened success of all tools on the simplest permutation types leads us to suggest that our participants’ scores did indeed reflect the quality of the word order. While not a strong conclusion, this allows us to place more weight on others drawn from the same information.

The second goal of the generation of automatic permutations was to provide insight into the evaluation techniques we have run on them. As discussed above, we believe we have a greater insight into the workings and priorities of machine translation evaluation systems. Our investigation is not exhaustive, however, and the reader is invited to draw their own additional conclusions if desired.

7.4.3 Relevance of structure to order evaluation

Thus far, we have responded to two of our central research questions. We have shown in Section 7.4.1 that one variant of DTED outperforms all baselines we used for comparison, but that only two tools were able to predict word-order judgments strictly better than they were able to predict holistic judgments. However, our investigation of structure merits greater investigation, as a result both of its significance in our metrics’ design stages and of its surprising *lack* of significance in our empirical work.

	Reference	Hypothesis
1	He cried for help to fight the bear off.	He bear while the resistance of the help for the noise.
2	The visiting team's only goal was scored by Memphis Depay.	The guests the only goal on the account of Memphis DEPA.
3	The cat sat on the mat.	The mat sat on the cat.

Table 7.5: Example word order mismatches

With a view to responding to our structure-related research question (Section 3.2.2), we have investigated the relevance of structure to the prediction of human opinions on the quality of adequacy in word order using two separate techniques. Represented by DTED and DERP, these represent a low-detail approach and a high-detail one. We have compared them with Kendall's τ, among other tools: an algorithm which ignores syntactic structure but otherwise evaluates in a very broadly similar manner to the other two tools.

The results of the investigations of these three tools, when compared together, lead us to a tentative conclusion: structure does not, in the cases of DTED and DERP, reliably lead to an improvement in the accuracy of predictions of word order quality. When meaningful syntactic structure is eliminated from the input provided to our tools, performance degrades dramatically in just two of five cases. In addition, in only one of five cases does a tool of ours noticeably outperform Kendall's τ when structure is included, a superiority which is retained when that tool ('c' DTED) is flattened.

We believe the lack of significant benefits afforded in our experiments by structured input to be a result of several factors. First, introducing structure necessarily also introduces noise to a sentence. This is because the question of parsing a sentence into an appropriate dependency structure is far from easy, and the tools we employ are not able to do so perfectly. This is at least in part due to the sheer complexities of the problem in question: as discussed by many others (see Chapter 2), natural languages have myriad edge cases, ambiguities, contextual variations and other complexities.

In addition to the inherent difficulties posed by the task of parsing, we believe that our tools may be inaccurate for an important other reason: their training datasets. In order to produce a dependency parser one must first train a model of the language being parsed, and for a number of reasons (see Section 3.4.3) we have used off-the-shelf systems. These are trained on English sentences which are intended to represent proper use of the language. For this reason, we expect them to perform well on proper English sentences, such as the human-produced reference sentences in WOJ-DB.

However, when faced with the many and varied flaws in machine-produced hypothesis sentences, such language models are likely to misfire. Partly this is unavoidable: even a human expert, faced with particularly low-quality translations such as hypothesis 1 in Table 7.5, may be unable to produce a meaningful structural description because there is simply none to find. Even in more reasonable cases such as hypothesis 2 in Table 7.5, tools trained on correct examples of English may produce an incorrect parse – here, for example, interpreting 'goal' as the principal verb rather than a dependent noun.

It is important to consider that while the parsers we used have been trained on proper sentences, this need not necessarily be the case. We have chosen the parsers and grammars which we have used for two reasons, described in Section 3.4.3: for convenience, and for their high quality. While these reasons are compelling, they are not absolute: in future, a parser could be trained on incorrect machine translations with parse trees written by human experts, potentially vastly reducing the noise in our parses.

A cheaper, if far less reliable, alternative to training a parser on a treebank of flawed sentences, it could be possible to integrate the relatively common concept of n-best parses. This relates to the methods by which parsers produce their trees, which in many cases is probabilistic. As mentioned in Section 2.2.3, this results in many alternative parses, each with an associated probability. In theory these could be extracted, as has been done with tools such as those of Owczarzak *et al.* [2007a,b], and either DTED or DERP run on many alternate parses: the highest of the resulting scores would then be assumed to best represent the sentence's true quality.

Short of producing such a specially-trained parser or extending our algorithms to support n-best parses, it would be possible to gain understanding of the effects of noise by measuring the quality of the parses which have been used in our experiments. Were we to be able to quantify their correctness, we could perform various comparative analyses to measure the relevance of that factor to the correlations we observe.

An analysis of parse quality would resolve the greatest problem with our identification of noise as the reason for our structured tools' relatively low success, namely that there is little evidence to unambiguously support it. However, generation of such analysis highlights the main problem with any truly reliable machine translation evaluation: it would need to be performed by human experts, requiring vastly more resources than were available for our project or may be available to many others.

Even if this were done, however – if we were able to guarantee that a given dependency parse was correct, minimising the negative effects of its use – it is still possible that its positive effects may be inconsequential. For example, while we believe it likely that a parser will correctly interpret sentences 3 in Table 7.5, the additional information provided by such a parse is unlikely to contain any insights into the sentence which would permit any structure-based algorithm from producing a *more* appropriate score than a structure-free tool like Kendall's τ.

Despite the various reasons to discount structure as a useful aid when evaluating translations, we do not believe our investigation is conclusive in that regard. Recall that while DTED performed almost equally well with and without structure, the same is not true of DERP, which achieved a markedly higher correlation with human judgments when provided with structure. While its more complex processing may have suffered more from negative factors such as low parse quality, causing worse results than DTED even with this greater information, we believe that higher quality inputs, coupled with DERP's detailed analysis, could lead to potentially dramatic improvements in performance.

Between the various reasons to expect noise in our parses, in addition to the occasionally negligible improvement permitted by even a correct parse, in retrospect we do not consider it surprising that structure affords only minor benefits in our experiments. However, given the limitations of our experiment – notably using parsers trained on correct sentences only – we consider that the benefit we observe, while small, is nonetheless important. We have observed empirically that structure *can* help in the evaluation of machine translation and we believe that that the full potential benefit, far from being limited to the effects observed in our experiments, could be vastly greater.

Chapter Eight

Conclusions

8.1 Existing work

In the field of machine translation, a number of approaches have been taken to evaluating the quality of produced translations, and in turn to evaluating those evaluation measures. A large number of these rely on variations on n-gram comparison using precision and recall, with metrics adding stemming, weighting, synonymy, probabilities and many other features.

Such n-grams can be extracted not from words alone, but from structural representations of sentences. A number of forms of dependency and other parsing produce tree structures, of which features such as relation labels can be compared with precision and recall, among other techniques.

Other evaluation approaches involve counting the number of changes which must be made to a translation to match a human equivalent. Again, a number of variations on this principle have been put forward, with differing ways of determining word matches and various approaches to the consideration of word ordering, such as low-cost block shifting operations.

This last evaluation element, the judgment of word order quality specifically, has recently gained focus as a discrete question. This is primarily in the context of other error types, with several projects attempting to provide granular overviews of the types and frequencies of errors across bodies of translated text. Word ordering, highlighted by several studies as one of the most detrimental error types in a number of ways, has received some attention through tools intended to judge it in isolation.

In order to judge the effectiveness of evaluation tools, they are compared with a gold standard: human quality judgments. These can take many forms, of which the most common is relative ranking of small groups of translations produced by different systems. Others involve absolute judgments of adequacy and/or fluency: the extent to which a given translation represents either the meaning of the original sentence or appropriate use of the target language.

In the wake of more granular automatic error analyses, the importance human evaluation of specific error types has begun to be recognised, with at least one dataset generated as part of the TaraXÜ project. However, this corpus does not include information detailed enough to provide in-depth analysis of metrics intended to measure individual error types including ordering.

8.2 Project overview

8.2.1 Research questions

This project is based on two assumptions: first, that the field of machine translation can benefit from a deeper automatically-generated yet accurate analysis of the quality of word ordering in existing systems; and second, that the structural information built into some non-granular metrics can help that evaluation.

To investigate these assumptions, we have asked three separate research questions (Section 3.2). The first such question we ask is whether by using structure we can produce state-of-the-art evaluation for word ordering. Secondly, we ask if the structure specifically contributes to the success (if any) of those evaluations. Finally, we query how relevant structure is to the evaluation of order in particular, compared with more holistic judgments.

8.2.2 Tools produced

In order to respond to these three questions, we have first produced two tools, each intended to evaluate word order quality in machine translation by taking advantage of structural information.

DTED **–** The first of these, DTED (Chapter 4), adapts the technique of Tree Edit Distance to the domain of translation evaluation. A Tree Edit Distance is the minimal count of atomic deletion, insertion and substitution operations on nodes required to convert one tree structure – a dependency parse – into another (Section 4.2.2). DTED's edit count is normalised to allow meaningful comparison between sentences.

Tree Edit Distance is intuitively related to the more commonly used technique of Levenshtein distances, making DTED a close cousin of various existing error-rate metrics such as Word Error Rate (Section 2.2.2). We consider DTED to be a direct combination of two existing techniques: dependency structure and error rate calculations.

The variations on DTED which we have produced (Section 4.3) represent different approaches to a key input: the alignments of words between sentences. With no clearly superior way of taking this feature into account, we have produced four separate versions of DTED which make greater or lesser use of it.

The simplest version, referred to as 'b' DTED and submitted to WMT 2016, considers nodes to be equal under no circumstances: all nodes must incur at least a substitution operation. This version performs the least well, contrary to 'c' DTED which uses externally produced alignment information to match related nodes at no cost, and which achieves the highest performance of all four variants.

The other two primary variants of DTED place a heavier emphasis on aligned words, by strictly prioritising any operations which affect such words. In the case of 'o' DTED, only these operations are considered when calculating a final score: any operations relating to unaligned nodes are ignored entirely. In 'l' DTED, such unaligned operations are assigned a weighting logarithmically proportional to the number of words in the sentences which are aligned. These variants of DTED perform well, but not quite to the level of 'c' DTED.

In order to observe the effect of structure in isolation, each variant of DTED has been run both on normal dependency structures and on 'flattened' structures in which the syntactic information has been replaced by purely linear relations between words (Sections 3.4.4, 4.4). To our surprise, this elimination of the information for which the tool was designed barely degrades its performance, with the correlations of only one variant, 'b' DTED, being strictly superior to those of its flattened equivalent (Section 7.3.2).

While DTED performs well, it has two major theoretical limitations: that it has little ability to judge the severity of any given error, and that it does not make any use of the labels indicating the nature of dependency links within the structures it is given.

DERP **–** Our second tool, DERP (Chapter 5), addresses these concerns by making active use of much more of the information contained within a dependency tree. Rather than considering operations on nodes alone, DERP compares the paths between nodes which are aligned between the two trees.

This is done through the calculation of Levenshtein distances on triples encoding three features of such paths (Section 5.3): the dependency labels they contain, their broad direction (right to left or left to right), and whether each edge is ascending or descending. As we have mentioned, Levenshtein distances and variants thereof are commonly used for machine translation evaluation, but they are generally applied to words – or, at the most complex level, to parts of speech and other annotations – rather than relations between those words.

These distances are combined to determine the minimum total edit distance such that every node is either directly or indirectly compared with every other (Section 5.6). This is done by calculating a minimum spanning tree in a meta-graph produced from the aligned nodes of one tree alone (Section 5.4). Distances are then normalised, as with DTED, to provide a score which can be reasonably compared between arbitrary sentences.

While the technique used by DERP to compare paths is usually associated with structure-free work, the observation of variations between the structural representations of words is related to the PEF-score of Stanojević and Sima'an [2014b]. However, while DERP calculates such disparity through differences in paths between nodes, PEF-score is calculated by summarising the disparity of an entire sentence at once. This is done by processing the various possible permutation trees [Gildea *et al.*, 2006] representing the difference in order of the various words, a purely mathematical process involving no syntactic or semantic information.

Interestingly, despite the increased depth of its analysis relative to DTED, DERP actually significantly underperforms at predicting human judgments of word order (Section 7.3.2). More in keeping with our expectations, the flattening of its input trees to minimise the structural information they contain does result in a noticeable drop in performance, suggesting that DERP makes positive use of that information.

8.2.3 Resources provided

WOJ-DB **–** The evaluation of metrics which themselves evaluate specific features of translation is not a trivial task. Given the lack of major resources providing scaled judgments on the quality of word order, we have produced our own: WOJ-DB (Chapter 6).

Our dataset was produced through a survey of diverse residents of our university town (Sections 6.3.1, 6.5), with sentence pairs generated from one of two sources. Half of the hypothesis sentences we use were generated by the diverse systems submitted to the Workshop on Machine Translation (Section 6.1.1), while the rest are randomly generated, through deliberately simplistic 'permutations', from WMT reference sentences in a manner not dissimilar to that of Kirchhoff *et al.* [2012].

Such automatic permutations represent a standalone contribution in our work, as discussed in Section 6.2.3. Split into four types, they represent varying numbers of predictable, simple changes to sentences, allowing us or others to inspect in isolation the effects of different facets of word ordering problems. Each reference sentence we chose was permuted from as little to as much as possible, with the sentence's $degree$ indicating the number of individual words affected.

The four permutation types are as follows. In 'order' permutations, $degree$ words are moved an ever greater distance from their starting position. In 'swap' permutations, $degree/2$ pairs of words whose parts of speech match have their positions inverted. In 'phrase' permutations, a random number of multi-word phrases containing a total of *degree* words are moved to a random phrase boundary elsewhere in the sentence. Finally, 'choice' permutations replace *degree* words in the base reference sentence by others with the same part of speech in a machine-produced hypothesis.

While permuted sentences are important on their own, the primary contribution of WOJ-DB is that of the judgments it contains. Participants were asked to rate each of 50 sentences according to two features: first, the overall adequacy (transference of meaning) of the hypothesis sentence; and second, the relevance of word ordering to that assessment (Section 6.2.1). Judgments were given on a 5-point Likert scale, with judgments gathered on 1783 sentence pairs overall.

We have analysed the contents of WOJ-DB in various ways (Section 7.2) to ensure that the judgments it contains follow expected trends, and that its composition does not allow any obvious bias. We are thus confident that it can be used for a wide variety of further projects as discussed in Section 8.4.

8.3 Conclusions

By conducting investigations through WOJ-DB to address the three research questions we introduced in Section 3.2, we have been able to evaluate DTED and DERP. The results of these investigations have been discussed in some detail in Section 7.4 and are summarised in Table 7.4. They lead us to a number of broad conclusions relating to the relevance of structure in evaluation both of word ordering, and of machine translation more generally.

8.3.1 Pursuing accurate evaluation

Our first goal was encompassed in the following question:

> Can we improve on the current state of the art of word order evaluation?

We believe we can answer this question in the affirmative, as one variant of DTED performs better than all the baselines we used, irrespective of the choice of third-party tools ('configuration') it relied on. This is a highly encouraging result, albeit one which is subject to an important caveat.

This caveat is that the baselines against which our tools were compared were chosen as representatives of different approaches, rather than because they themselves represented the cutting edge in order evaluation specifically. Thus, while the success of our tools in outperforming BLEU and Meteor is far from being uninteresting, a comparison against other tools may reveal limits to our tools' superiority. It should be noted that Kendall's τ, the only one of our baselines explicitly designed and used for order evaluation [Birch and Osborne, 2010; Isozaki *et al.*, 2010], was also the highest-performing competitor in our experiments.

Caveats aside, we consider the success of our tools at predicting human judgments to be a vindication of our techniques: an indication that DTED especially and, to a lesser extent, DERP can be a meaningful automatic descriptor of the quality of word ordering. They suggest that both node edit distances and more complex path differences may well be worthwhile components if built into future metrics: whether or not our metrics strictly outperform all existing metrics, the techniques they encompass can indeed push forward the state of the art.

8.3.2 Relevance of structure

Our second investigation was into the true relevance of the key component of our algorithms:

> Does structure aid the evaluation of word order in machine translation?

Our results were not entirely in keeping with our expectations, as several variants of DTED performed almost equally well when deprived of genuine structural information. Our experiments thus suggest that tree edit distance and simple Word Error Rate are approximately comparable in their prediction of word order quality: the grouping of words by relation affects evaluation neither positively nor negatively.

This is an intriguing result in itself, as a complete lack of effect is arguably more surprising even than structure damaging predictive ability. We believe our results to be a result of two conflicting factors: first, correct structural information improving the automatic assessments; and second, incorrect structure caused by flawed parses having the opposite effect. Such flawed parses are in some senses an inevitable consequence of the imperfections in machine-produced hypotheses, though it may be possible to address this through training grammars on such hypotheses.

While DTED was unable, for the most part, to improve its performance through the use of structural information, the same was not true with DERP. In the case of this more complex algorithm, the removal of structure resulted in at least an 8.8% reduction in correlation with human judgments. We consider this to indicate that while DTED makes little enough use of structural information to avoid being significantly affected by imperfections at the parsing level, DERP's use of such information is more intelligent and in-depth: while its performance is more damaged by negative factors, the information which is reliable is nonetheless put to good use.

The improvement in performance of our tools when provided with structural information – mild in the case of simpler DTED, but more marked in the case of more complex DERP – leads us to the conclusion that structure is indeed helpful to them. Negative factors, which may be related to the quality of hypothesis-sentence parses or other aspects of dependency parsing, appear to limit the ability of our tools to extract exclusively useful conclusions from that structure, but rather than discounting its utility, we believe this merely calls for further work in the area.

8.3.3 Cohesion of structure and order

Our third research question pertains not to structure, but to the feature of language we are investigating through it:

> Does dependency structure permit word order evaluation, or does it lend itself more to holistic judgment?

Our response to this question is somewhat obfuscated by a simple observation already presented in literature: word order is a powerful predictor of overall sentence quality in English, with ordering problems confusing overall meaning to the extent that overall clarity is difficult without at least mostly correct ordering. Given this, it is perhaps unsurprising that our tools performed in most cases approximately equally well when predicting holistic adequacy compared with word ordering specifically.

The exceptions to this trend were the two highest-performing variants of DTED, which observed a very limited drop in performance when considered in the more general context of adequacy, both with and without the structural information being removed from its input through flattening. We believe that this drop, while not as marked as we may have expected, mildly suggests the effect we expected: that DTED focuses more on order than holistic predictions.

It is relevant to note that while our tools observed no difference or a small decrease in performance when predicting holistic adequacy, the results of our baselines were much more dramatic. Kendall's τ, designed exclusively to measure order quality, performed 35.5% better at predicting such judgments than their more general alternatives; while Meteor, designed for holistic judgments, correlated with holistic assessments 55.9% better than with those of order.

Our response to the research question is thus a tentative one: we believe that structure is far from useless in the evaluation of word ordering, though our results suggest that other factors are assessed alongside this. We can assume these factors to be the syntactic ones, such as word choice and form, which play a major role when producing parse trees. Thus, our approaches represent interesting middle grounds between the evaluation of specific and general quality assessments.

8.4 Future work

While each element of our investigations has produced results which are interesting in their own right, they are necessarily incomplete. As discussed in Section 3.3, we have imposed a number of practical limitations to our project due simply to the limited resources available. Further to these, the conclusions we have drawn have raised further questions themselves.

Semi-automation **–** Arguably the most central question to our investigation of structure is to do with the true potential applicability of this feature. While we believe that the clear success of DTED, and the more limited success of DERP, show that dependency structure comparison is a worthwhile item in the machine translation evaluation toolbelt, its true utility is unplumbed. We recommend the development of an equivalent metric relying in part on human input: for example, DERP run on human-generated reference and hypothesis dependency trees.

This approach has already been taken with semi-automatic tools such as HTER [Snover *et al.*, 2006] and HMEANT [Lo and Wu, 2011]. The experimental results of such tools were reported as a way of gauging the relevance of the appropriate techniques, which were additionally packaged in a fully automated form as TER and MEANT respectively.

Tailored training **–** While human involvement in generating parse trees would provide the immediate and generic benefit of a greater understanding of the potential help of structure, it might also permit a more long-lasting and specific benefit: the development of more appropriately trained parsers.

As discussed in Section 2.1.3, producing a parser which can be applied to a specific language generally requires a large number of sentences in that language. For our experiments, we have used tools trained on correct English (Section 3.4.3), although in practice half the sentences we encounter are potentially flawed machine translations.

Thus, were a treebank to be produced containing human-parsed machine-produced dependency structures, such a resource could be used to train future dependency parsers, which in turn could be used by tools similar to ours. The presumably more accurate resulting parse trees could be expected to result in higher-accuracy evaluations.

Single-tree comparisons **–** While flawed hypothesis sentences are likely to lead to even more flawed hypothesis parses without closely tailored training data, an alternative approach to structural evaluation is to avoid hypothesis parsing and generate structures for the reference sentence only. This is the approach taken by the BLEUÂTRE [Mehay and Brew, 2006] and RED [Yu *et al.*, 2014] metrics (see Section 2.2.3), which extract word n-grams from a reference tree before running traditional comparison techniques relating to precision and recall.

An adaptation of this approach to DERP could involve comparisons of dependency path lengths between two aligned reference words to the flat word-based distances between matching hypothesis words. Alternatively, errors could be measured by calculating the pathwise moves or swaps which would be required in a reference tree for its nodes to be in the same order as the matching words in a hypothesis sentence. Projectivity could be imposed as an additional constraint to prevent trivial solutions.

Given the necessarily asymmetric and limited information available to any algorithm relying on a single dependency tree, the relevance of any such approach would be far from guaranteed. However, we believe that such approaches could lead to a 'best of both worlds' solution to the evaluation problems we have investigated, making use of helpful structural information while omitting anything unhelpful.

Tool publications **–** While various adaptations could be made to our tools to improve their performance, they nonetheless already represent both novel and largely successful approaches to the question of order evaluation. As such, it may be of benefit to the machine translation community to make up to three separate tools publically available for download.

We believe that implementations of both DTED and DERP can be of direct use when training translation systems. The surprising success of flattened 'c' DTED, coupled with the obvious possible execution-speed gains possible by omitting its parsing phase, suggests that it may be at least as helpful as its structured counterparts.

Restriction removal **–** Given the general success of our tools, and the range of questions raised by their performance, we feel that it would be worthwhile to continue our investigations through the removal of the restrictions we imposed in Section 3.3. These were the focus on translations into English only, the investigation of adequacy over fluency, and the calculation of scores for sentence pairs only rather than whole systems or hypotheses with multiple references.

Each of these limitations was imposed for purely practical reasons, and could potentially be lifted without excessive amounts of extra work. Given the lack of language-specific features in our algorithms, and our deliberate avoidance of reliance on a single configuration of third-party tools, the tools we have produced could be adapted to other languages by simply providing suitable non-English training corpora, or indeed pre-trained parsers, taggers and alignment generators.

The generation of system-level and multi-reference scores has already been discussed in Section 3.3: these could be done by simple arithmetic averaging in the former case, and selecting the highest pairwise score out of several in the latter.

The evaluation of fluency would be a more nebulous change, as nothing in our tools precludes them from fluency evaluation in their current form. Instead of altering the tools themselves, we suggest an extension to WOJ-DB – another survey – with questions relating to grammatical correctness rather than transfer of meaning.

***WOJ-DB applications* –** Just as our tools could be improved yet already represent worthwhile forays into evaluation using structure, so WOJ-DB, while limited to adequacy, can be a powerful resource. We have already shown its utility when applied to the evaluation of our tools, but have only scratched the surface of the knowledge it can provide on other tools.

The most notable other metrics whose evaluation would be helpful to consider would be those which judge order specifically, as introduced in Section 2.3.1. Evaluations of all such tools would provide interesting comparisons to our own tools, allowing more broad evaluation than the few prominent baselines we have included.

We would primarily recommend the more detailed evaluation of the tools whose WMT data performed relatively well already. Notable among these are the multi-approach and recently highly lauded BEER metric [Stanojević and Sima'an, 2014a], which includes the standalone order component proposed by Stanojević and Sima'an [2014b]. Results relating to the DiscoTK metric [Joty *et al.*, 2014], though not statistically significant in our experiments, are also encouraging to the point that further investigation is warranted.

The uses of WOJ-DB are not inherently limited to metric evaluation. We believe the two types of human judgments we have gathered can be used to gain a deeper understanding of the relevance of order to overall quality, perhaps through deeper analysis of the sentences we have included. Of these, Section 7.4.2 represents only an initial look into the implications of our automatic permutations: the precise information they contain about error types could yield numerous insights if compared between many different tools, compared with different qualities or types of translated sentences, or investigated in the context of individual participants' backgrounds.

While such further investigations into WOJ-DB, DTED and DERP were not considered essential to our own research questions, they nonetheless have the potential to deepen our understanding of error behaviour and impact, the relevance of structure to evaluation of both granular and general scores, and the specific improvements needed for individual translation tools.

BIBLIOGRAPHY

Akiba, Y., Imamura, K., and Sumita, E. (2001). Using multiple edit distances to automatically rank machine translation output. In Proceedings of the MT Summit VIII.

Al-Onaizan, Y., Curin, J., Jahr, M., Knight, K., Lafferty, J. D., Melamed, D., Och, F. J., Purdy, D., Smith, N. A., and Yarowsky, D. (1999). Statistical machine translation. Technical report, Final Report, JHU Summer Workshop.

Amazon (2005). Amazon Mechanical Turk. https://www.mturk.com/. Last accessed on 15 March 2017.

Amigó, E., Giménez, J., Gonzalo, J., and Màrquez, L. (2006). MT Evaluation: Human-like vs. Human Acceptable. In Proceedings of the International Conference on Computational Linguistics and the Association for Computational Linguistics, pages 17–24, Sydney. Association for Computational Linguistics.

Avramidis, E., Burchardt, A., Federmann, C., Popović, M., Tscherwinka, C., and Vilar, D. (2012). Involving Language Professionals in the Evaluation of Machine Translation. In Proceedings of the International Conference on Language Resources and Evaluation (LREC), pages 1127–1130.

Avramidis, E., Burchardt, A., Hunsicker, S., Popović, M., Tscherwinka, C., Vilar, D., and Uszkoreit, H. (2014). The TaraXÜ Corpus of Human-Annotated Machine Translations. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC), pages 2679–2682, Reykjavik, Iceland. European Language Resources Association (ELRA).

Babych, B. and Hartley, A. (2004). Extending the BLEU MT evaluation method with frequency weightings. In Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, page 621, Morristown, NJ, USA. Association for Computational Linguistics.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. In Proceedings of the International Conference on Learning Representations, pages 1–15.

Banerjee, S. and Lavie, A. (2005). METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pages 65–72. Association for Computational Linguistics.

Berg, B. (2015). SDAPS. http://sdaps.org/. Last accessed on 1 September 2016.

Berka, J., Bojar, O., Fishel, M., Popović, M., and Zeman, D. (2012). Automatic MT Error Analysis: Hjerson Helping Addicter. In Proceedings of the International Conference on Language Resources and Evaluation, pages 2158–2163.

Bhate, S. and Kak, S. (1991). Panini's Grammar and Computer Science. Annals of the Bhandarkar Oriental Research Institute, 72(1):79–94.

Bille, P. (2005). A survey on tree edit distance and related problems. Theoretical Computer Science, 337(1-3):217–239.

Birch, A. and Osborne, M. (2010). LRscore for evaluating lexical and reordering quality in MT. In Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and Metrics, pages 327–332.

Birch, A., Osborne, M., and Blunsom, P. (2010). Metrics for MT evaluation: evaluating reordering. Machine Translation, 24(1):15–26.

Birch, A., Osborne, M., and Koehn, P. (2008). Predicting success in machine translation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 745–754.

Bird, S. (2006). NLTK: The Natural Language Toolkit. In Proceedings of the COLING/ACL on Interactive presentation sessions (COLING-ACL '06), pages 69–72.

Bojar, O., Buck, C., Federmann, C., Haddow, B., Koehn, P., Leveling, J., Monz, C., Pecina, P., Post, M., Saint-Amand, H., Soricut, R., Specia, L., and Tamchyna, A. (2014). Findings of the 2014 Workshop on Statistical Machine Translation. In Proceedings of the Ninth Workshop on Statistical Machine Translation, pages 12–58.

Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huck, M., Jimeno Yepes, A., Koehn, P., Logacheva, V., Monz, C., Negri, M., Névéol, A., Neves, M., Popel, M., Post, M., Rubino, R., Scarton, C., Specia, L., Turchi, M., Verspoor, K., and Zampieri, M. (2016a). Findings of the 2016 Conference on Machine Translation. In Proceedings of the First Conference on Machine Translation, volume 2, pages 131–198.

Bojar, O., Chatterjee, R., Federmann, C., Haddow, B., Huck, M., Hokamp, C., Koehn, P., Logacheva, V., Monz, C., Negri, M., Post, M., Scarton, C., Specia, L., and Turchi, M. (2015). Findings of the 2015 Workshop on Statistical Machine Translation. In Proceedings of the 10th Workshop on Statistical Machine Translation, pages 1–46, Lisboa, Portugal.

Bojar, O., Graham, Y., Kamran, A., and Stanojević, M. (2016b). Results of the WMT16 Metrics Shared Task. In Proceedings of the First Conference on Machine Translation, volume 2, pages 199–231, Berlin, Germany.

Brill, E. (1992). A Simple Rule-Based Part of Speech Tagger. In Proceedings of the Workshop on Speech and Natural Language, pages 112–116.

Brown, P. F., Cocke, J., della Pietra, S. A., della Pietra, V. J., Jelinek, F., Lafferty, J. D., Mercer, R. L., and Roossin, P. S. (1990). A statistical approach to machine translation. Computational Linguistics, 16(2):79–85.

Brown, P. F., Della Pietra, S. A., della Pietra, V. J., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. Computational Linguistics, 10598.

Brown, P. F., Lai, J. C., and Mercer, R. L. (1991). Aligning Sentences in Parallel Corpora. In Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics, pages 169–176.

Büchse, M., Nederhof, M.-J., and Vogler, H. (2011). Tree parsing with synchronous tree-adjoining grammars. In Proceedings of the 12th International Conference on Parsing Technologies, pages 14–25.

Cahill, A., Burke, M., O'Donovan, R., van Genabith, J., and Way, A. (2004). Long-Distance Dependency Resolution in Automatically Acquired Wide-Coverage PCFG-Based LFG Approximations. In Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, pages 319–326.

Callison-Burch, C., Fordyce, C., Koehn, P., Monz, C., and Schroeder, J. (2007). mbox(Meta-) Evaluation of Machine Translation. In Proceedings of the Second Workshop on Statistical Machine Translation, pages 136–158.

Cer, D. M., de Marneffe, M.-C., Jurafsky, D., and Manning, C. D. (2010). Parsing to Stanford Dependencies: Trade-offs between Speed and Accuracy. In Proceedings of the Seventh Conference on International Language Resources and Evaluation.

Chan, Y. S., Ng, H. T., and Link, L. (2008). MAXSIM: A Maximum Similarity Metric for Machine Translation Evaluation. In Proceedings of ACL-08: HLT, pages 55–62. Association for Computational Linguistics.

Chen, B. and Kuhn, R. (2011). Amber: A modified BLEU, enhanced ranking metric. In Proceedings of the 6th Workshop on Statistical Machine Translation, pages 71–77.

Chen, J. and Nie, J.-Y. (2000). Automatic construction of parallel English-Chinese corpus for cross-language information retrieval. In Proceedings of the Sixth Conference on Applied Natural Language Processing, pages 21–28. Association for Computational Linguistics.

Chiang, D., Andreas, J., Bauer, D., Hermann, K. M., Jones, B. K., and Knight, K. (2013). Parsing Graphs with Hyperedge Replacement Grammars. In Proceedings of the 51st Meeting of the ACL.

Chiang, D. and Knight, K. (2006). An introduction to synchronous grammars. Tutorial available at http://www.isi.edu/˜chiang/papers/synchtut.pdf.

Chow, S., Shao, J., and Wang, H. (2008). Sample Size Calculations in Clinical Research. Chapman & Hall/CRC Biostatistics Series, second edition.

Clark, A. (2003). Pre-processing very noisy text. In Proceedings of Workshop on Shallow Processing of Large Corpora, pages 12–22.

Clark, S., Hockenmaier, J., and Steedman, M. (2002). Building deep dependency structures with a wide-coverage CCG parser. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pages 327–334.

Čmejrek, M., Cuřín, J., Havelka, J., Hajič, J., and KuboÅĹ, V. (2004). Prague Czech-English Dependency Treebank: Syntactically Annotated Resources for Machine Translation. In Proceedings of the 4th International Conference on Language Resources and Evaluation, page 4, Lisbon, Portugal.

Collins, M. and Duffy, N. (2001). Convolution kernels for natural language. Advances in Neural Information Processing Systems, 14:625–632.

Comelles, E. and Atserias, J. (2015). VERTa: a Linguistically-motivated Metric at the WMT15 Metrics Task. In Proceedings of the Tenth Workshop on Statistical Machine Translation, pages 366–372.

Dagan, I., Glickman, O., and Magnini, B. (2006). The PASCAL recognising textual entailment challenge. Machine Learning Challenges, page 177.

de Marneffe, M.-C., MacCartney, B., and Manning, C. D. (2006). Generating typed dependency parses from phrase structure parses. In Proceedings of the 5th International Conference on Language Resources and Evaluation, pages 449–454.

Demaine, E. D., Mozes, S., Rossman, B., and Weimann, O. (2009). An optimal decomposition algorithm for tree edit distance. ACM Transactions on Algorithms, 6:1–19.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological), 39(1):1–38.

DeNeefe, S. and Knight, K. (2009). Synchronous tree adjoining machine translation. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, volume 2, pages 727–736.

Denkowski, M. J. and Lavie, A. (2010). Choosing the Right Evaluation for Machine Translation: an Examination of Annotator and Automatic Metric Performance on Human Judgment Tasks. In Proceedings of the Ninth Biennial Conference of the Association for Machine Translation in the Americas.

Denkowski, M. J. and Lavie, A. (2011). Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In Proceedings of the Sixth Workshop on Statistical Machine Translation, pages 85–91.

Denkowski, M. J. and Lavie, A. (2014). Meteor Universal: Language Specific Translation Evaluation for Any Target Language. In Proceedings of the EACL 2014 Workshop on Statistical Machine Translation.

Ding, Y. and Palmer, M. (2005). Machine translation using probabilistic synchronous dependency insertion grammars. In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pages 541–548.

Doddington, G. (2002). Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In Proceedings of the Second International Conference on Human Language Technology Research, pages 138–145.

Druck, G., Mann, G., and McCallum, A. (2009). Semi-supervised Learning of Dependency Parsers using Generalized Expectation Criteria. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, pages 360–368.

Durrani, N., Haddow, B., Koehn, P., and Heafield, K. (2014). Edinburgh's Phrase-based Machine Translation Systems for WMT-14. In Proceedings of the Ninth Workshop on Statistical Machine Translation, pages 97–104, Baltimore, Maryland USA. Association for Computational Linguistics.

Dyer, C. (2010). Two monolingual parses are better than one (synchronous parse). In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 263–266. Association for Computational Linguistics.

Dyer, C., Lopez, A., Ganitkevitch, J., Weese, J., Ture, F., Blunsom, P., Setiawan, H., Eidelman, V., and Resnik, P. (2010). cdec: A Decoder, Alignment, and Learning Framework for Finite-State and Context-Free Translation Models. In Proceedings of the ACL 2010 System Demonstrations, pages 7–12.

Earley, J. (1970). An efficient context-free parsing algorithm. Communications of the ACM, 13(2):94–102.

El Kholy, A. and Habash, N. (2011). Automatic Error Analysis for Morphologically Rich Languages. In MT Summit XIII, pages 225–232, Xiamen, China.

Federmann, C. (2012). Appraise: an Open-Source Toolkit for Manual Evaluation of MT Output. The Prague Bulletin of Mathematical Linguistics, 98:25–35.

Fishel, M., Bojar, O., and Popović, M. (2012a). Terra: a Collection of Translation Error-Annotated Corpora. In Proceedings of the 8th International Conference on Language Resources and Evaluation, pages 7–14, Istanbul, Turkey.

Fishel, M., Sennrich, R., Popović, M., and Bojar, O. (2012b). TerrorCat: a Translation Error Categorization-based MT Quality Metric. In Proceedings of the 7th Workshop on Statistical Machine Translation, pages 64–70, Montréal, Canada. Association for Computational Linguistics.

Flanagan, M. A. (1994). Error Classification for MT Evaluation. In Technology Partnerships for Crossing the Language Barrier: Proceedings of the First Conference of the Association for Machine Translation in the Americas, pages 65–72.

Fomicheva, M. and Specia, L. (2016). Reference Bias in Monolingual Machine Translation Evaluation. In The 54th Annual Meeting of the Association for Computational Linguistics: Proceedings of the Conference, volume 1, pages 77–82.

Font Llitjós, A., Carbonell, J. G., and Lavie, A. (2005). A Framework for Interactive and Automatic Refinement of Transfer-based Machine Translation. In Proceedings of the Tenth Workshop of the European Association for Machine Translation (EAMT).

Forcada, M. L., Ginestí-Rosell, M., Nordfalk, J., O'Regan, J., Ortiz-Rjojas, S., Pérez-Ortiz, J. A., Sánchez-Martínez, F., Ramírez-Sánchez, G., and Tyers, F. M. (2011). Apertium: a free/open-source platform for rule-based machine translation. Machine Translation, 25(2):127–144.

Francis, W. N. (1964). A Standard Sample of Present-Day English for Use with Digital Computers. Technical report, Report to the U.S Office of Education on Cooperative Research Project No. E-007.

Fung, P. and Church, K. W. (1994). K-vec: A new approach for aligning parallel texts. In Proceedings of the 15th Conference on Computational Linguistics: Volume 2, pages 1096–1102.

Gaifman, H. (1965). Dependency systems and phrase-structure systems. Information and Control, 8(3):304–337.

Galley, M. and Manning, C. D. (2009). Quadratic-time dependency parsing for machine translation. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2, pages 773–781. Association for Computational Linguistics.

Galton, F. (1886). Regression Towards Mediocrity in Hereditary Stature. http://www.jstor.org/stable/2841583?origin=crossref.

Gamon, M., Aue, A., and Smets, M. (2005). Sentence-level MT evaluation without reference translations: Beyond language modeling. In Proceedings of the European Association for Machine Translation, pages 103–111.

Gildea, D., Satta, G., and Zhang, H. (2006). Factoring Synchronous Grammars by Sorting. In Proceedings of the International Conference on Computational Linguistics and the Association for Computational Linguistics, pages 279–286.

Giménez, J. and Màrquez, L. (2007). Linguistic Features for Automatic Evaluation of Heterogeneous MT Systems. In Proceedings of the Second ACL Workshop on Statistical Machine Translation, pages 256–264, Prague, Czech Republic. Association for Computational Linguistics.

Gimpel, K. and Smith, N. A. (2011). Quasi-synchronous phrase dependency grammars for machine translation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, pages 474–485.

Gómez-Rodríguez, C., Carroll, J., and Weir, D. J. (2011). Dependency parsing schemata and mildly non-projective dependency parsing. Computational Linguistics, 37(3):541–586.

Goto, I., Lu, B., Chow, K. P., Sumita, E., Tsou, B. K., Utiyama, M., and Yasuda, K. (2013). Database of Human Evaluations of Machine Translation. Journal of Natural Language Processing, 20(1):27–57.

Graehl, J., Knight, K., and May, J. (2004). Training Tree Transducers. Technical Report October 2003, DTIC Document.

Grefenstette, G. and Tapanainen, P. (1994). What is a word, What is a sentence? Problems of tokenization. In COMPLEX 1994: Proceedings of the 3rd Conference on Computational Lexicography and Text Research, pages 79–87, Budapest, Hungary.

Guzmán, F., Joty, S., Màrquez, L., and Nakov, P. (2014). Using Discourse Structure Improves Machine Translation Evaluation. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, volume 1, pages 687–698.

Guzmán, F., Joty, S., Màrquez, L., and Nakov, P. (2017). Machine Translation Evaluation with Neural Networks. Computer Speech & Language, 45:180–200.

Habash, N. and Elkholy, A. (2008). SEPIA: Surface Span Extension to Syntactic Dependency Precision-based MT Evaluation. In Proceedings of the NIST Metrics for Machine Translation Workshop at the Association for Machine Translation in the Americas Conference, Waikiki, HI.

Hadiwinoto, C. and Ng, H. T. (2017). A Dependency-Based Neural Reordering Model for Statistical Machine Translation. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, pages 109–115.

Hajič, J., Hajičová, E., Panevová, J., Sgall, P., Bojar, O., Cinková, S., Fučíková, E., Mikulová, M., Pajas, P., Popelka, J., Semecký, J., Šindlerová, J., Štěpánek, J., Toman, J., Urešová, Z., and Žabokrtský, Z. (2012). Announcing Prague Czech-English Dependency Treebank 2.0. In Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC'12), pages 3153–3160.

Hamming, R. W. (1950). Error Detecting and Error Correcting Codes.

Hays, D. G. (1962). Automatic language-data processing. Computer Applications in the Behavioral Sciences, pages 394–423.

He, Y. (2010). The DCU Dependency-Based Metric in WMT-MetricsMATR 2010. In Proceedings of the Joint 5th Workshop on Statistical Machine Translation and MetricsMATR, pages 349–353, Uppsala, Sweden.

Higher Education Statistics Agency (2016). HE students by HE provider, level of study, mode of study and domicile. Technical report, Higher Education Statistics Agency.

Hockenmaier, J. and Steedman, M. (2007). CCGbank: A Corpus of CCG Derivations and Dependency Structures Extracted from the Penn Treebank. Computational Linguistics, 33(3):355–396.

Hopcroft, J. E., Motwani, R., and Ullman, J. D. (2001). Introduction to automata theory, languages, and computation. ACM SIGACT News, 32(1):60–65.

Isozaki, H., Hirao, T., Duh, K., Sudoh, K., and Tsukada, H. (2010). Automatic Evaluation of Translation Quality for Distant Language Pairs. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 944–952.

Jean, S., Firat, O., Cho, K., Memisevic, R., and Bengio, Y. (2015). Montreal Neural Machine Translation Systems for WMT15. In Proceedings of the Tenth Workshop on Statistical Machine Translation, pages 134–140.

Johnson, R., King, M., and des Tombe, L. (1985). Eurotra: a Multilingual System under Development. Computational Linguistics, 11(2-3):155–169.

Joshi, A. K., Levy, L. S., and Takahashi, M. (1975). Tree adjunct grammars. Journal of computer and system sciences, pages 136–163.

Joty, S., Carenini, G., and Ng, R. T. (2012). A Novel Discriminative Framework for Sentence-Level Discourse Analysis. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pages 904–915.

Joty, S., Guzmán, F., Màrquez, L., and Nakov, P. (2014). DiscoTK: Using Discourse Structure for Machine Translation Evaluation. In Proceedings of the Ninth Workshop on Statistical Machine Translation, pages 402–408.

Kahn, J. G., Snover, M., and Ostendorf, M. (2009). Expected Dependency Pair Match: Predicting translation quality with expected syntactic structure. Machine Translation, 23(2-3):169–179.

Kallmeyer, L. (2010). Parsing beyond context-free grammars. Springer.

Kallmeyer, L. (2013). Linear Context-Free Rewriting Systems. Language and Linguistics Compass, 7(1):22–38.

Kendall, M. G. (1938). A New Measure of Rank Correlation. Biometrika, 30(1):81–93.

Kingsbury, P. and Palmer, M. (2002). From TreeBank to PropBank. In Proceedings of the International Conference on Language Resources and Evaluation, pages 1989–1993.

Kirchhoff, K., Capurro, D., and Turner, A. (2012). Evaluating User Preferences in Machine Translation Using Conjoint Analysis. In EAMT 2012: Proceedings of the 16th Annual Conference of the European Association for Machine Translation, pages 119–126, Trento, Italy.

Klein, D. and Manning, C. D. (2003a). A* parsing: fast exact Viterbi parse selection. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Volume 1, pages 40–47. Association for Computational Linguistics.

Klein, D. and Manning, C. D. (2003b). Accurate Unlexicalized Parsing. In Proceedings of the 41st Meeting of the Association for Computational Linguistics, pages 423–430. Association for Computational Linguistics.

Klein, D. and Manning, C. D. (2004). Corpus-based induction of syntactic structure: models of dependency and constituency. In Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, volume 1, pages 478–485.

Klein, S. and Simmons, R. F. (1963). A computational approach to grammatical coding of English words. Journal of the ACM, 10(3):334–347.

Knuth, D. E. (1965). On the translation of languages from left to right. Information and Control, 8(6):607–639.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. MT summit, 11:79–86.

Koehn, P. (2010). Statistical machine translation. Cambridge University Press.

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, pages 177–180. Association for Computational Linguistics.

Koehn, P. and Monz, C. (2006). Manual and Automatic Evaluation of Machine Translation between European Languages. In Proceedings of the Workshop on Statistical Machine Translation, pages 102–121.

Koehn, P. and Schroeder, J. (2011). European Parliament Proceedings Parallel Corpus. http://www.statmt.org/europarl/archives.html#v6. Last accessed on 2 February 2017.

Kruskal, J. B. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem. Proceedings of the American Mathematical Society, 7(1):48–50.

Kwiatkowski, T., Zettlemoyer, L., Goldwater, S., and Steedman, M. (2010). Inducing Probabilistic CCG Grammars from Logical Form with Higher-Order Unification. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, pages 1223–1233. Association for Computational Linguistics.

Laird, N. M. and Ware, J. H. (1982). Random-Effects Models for Longitudinal Data. Biometrics, 38(4):963–974.

Lapata, M. (2003). Probabilistic text structuring: Experiments with sentence ordering. In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, pages 545–552.

Lapata, M. (2006). Automatic Evaluation of Information Ordering: Kendall's Tau. Computational Linguistics, 32(4):471–484.

Lari, K. and Young, S. J. (1990). The estimation of stochastic context-free grammars using the Inside-Outside algorithm. Computer Speech and Language, 4(1):35–56.

Lavie, A. and Agarwal, A. (2007). METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In Proceedings of the Second Workshop on Statistical Machine Translation, pages 228–231.

Lavie, A. and Denkowski, M. J. (2009). The Meteor metric for automatic evaluation of machine translation. Machine Translation, 23(2-3):105–115.

Lavie, A., Sagae, K., and Jayaraman, S. (2004). The Significance of Recall in Automatic Metrics for MT Evaluation. In Proceedings of the 6th Conference of the Association for Machine Translation in the Americas, pages 134–143.

Leusch, G., Ueffing, N., and Ney, H. (2006). CDer: Efficient MT evaluation using block movements. In Proceedings of EACL-2006 (11th Conference of the European Chapter of the Association for Computational Linguistics), pages 241–248.

Levenshtein, V. I. (1965). Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady, 10(1):707–710.

Liang, P., Taskar, B., and Klein, D. (2006). Alignment by agreement. In Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, pages 104–111.

Likert, R. (1932). A Technique for the Measurement of Attitudes. Archives of Psychology, 140:1–55.

Linguistic Data Consortium (2011). 2008/2010 NIST Metrics for Machine Translation (MetricsMaTr) GALE Evaluation Set LDC2011T05. https://catalog.ldc.upenn.edu/LDC2011T05. Last accessed on 20 Aug 2015.

Liu, D. and Gildea, D. (2005). Syntactic Features for Evaluation of Machine Translation. In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, pages 25–32.

Liu, S., Li, S., Zhao, T., and Li, S. (2010). A Dependency-Based Neural Reordering Model for Statistical Machine Translation. In Proceedings of the 6th International Conference on Natural Language Processing and Knowledge Engineering, pages 1–6.

Lo, C.-k., Tumuluru, A. K., and Wu, D. (2012). Fully automatic semantic MT evaluation. In Proceedings of the Seventh Workshop on Statistical Machine Translation, pages 243–252.

Lo, C.-k. and Wu, D. (2011). MEANT: An Inexpensive, High-accuracy, Semi-automatic Metric for Evaluating Translation Utility via Semantic Frames. In Proceedings of ACL-HLT, pages 220–229.

Lommel, A. R., Burchardt, A., Popović, M., Harris, K., Avramidis, E., and Uszkoreit, H. (2014). Using a New Analytic Measure for the Annotation and Analysis of MT Errors on Real Data. In Proceedings of the 17th Annual Conference of the European Association for Machine Translation, pages 165–172, Dubrovnik, Croatia. European Association for Machine Translation.

Lopez, A. (2008). Statistical machine translation. ACM Computing Surveys, 40(3):1–49.

Ma, Q., Graham, Y., Baldwin, T., and Liu, Q. (2017). Further Investigation into Reference Bias in Monolingual Evaluation of Machine Translation. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2466–2475.

Macháček, M. and Bojar, O. (2013). Results of the WMT13 metrics shared task. In Proceedings of the Eighth Workshop on Statistical Machine Translation, pages 43–49.

Malecha, G. and Smith, I. (2010). Maximum Entropy Part-of-Speech Tagging in NLTK. Unpublished course-related report.

Manning, C. D. and Schütze, H. (1999). Foundations of Statistical Natural Language Processing. MIT Press.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., and McClosky, D. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. In Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pages 55–60.

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. (1993). Building a large annotated corpus of English: The Penn Treebank. Computational Linguistics.

Marzal, A. and Vidal, E. (1995). Fast Computation of Normalized Edit Distances. IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(8):899–902.

McCaffery, M. and Nederhof, M.-J. (2016). DTED: Evaluation of Machine Translation Structure Using Dependency Parsing and Tree Edit Distance. In Proceedings of the First Conference on Machine Translation, volume 2, pages 491–498, Berlin, Germany.

Mehay, D. N. and Brew, C. (2006). BLEUÂTRE: Flattening Syntactic Dependencies for MT Evaluation. In Proceedings of MT Summit XII, pages 122–131.

Melamed, I. D. (2000). Models of translational equivalence among words. Computational Linguistics, 26(2):221–249.

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.

Miller, G. A. (1995). WordNet: a lexical database for English. Communications of the ACM, 38(11):39–41.

Moulton, B. R. (1986). Random Group Effects and the Precision of Estimates. Journal of Econometrics, 32:385–397.

Nadkarni, P. M., Ohno-Machado, L., and Chapman, W. W. (2011). Natural language processing: an introduction. Journal of the American Medical Informatics Association, 18(5):544–51.

Necas, D. and Haapala, A. (2014). Python-Levenshtein 0.12.0. https://pypi.python.org/pypi/python-Levenshtein/0.12.0. Last accessed on 12 March 2017.

Nederhof, M.-J. and Satta, G. (2004). An alternative method of training probabilistic LR parsers. In Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, page 550.

Nederhof, M.-J. and Satta, G. (2006). Probabilistic parsing strategies. Journal of the ACM.

Nederhof, M.-J. and Satta, G. (2008). Probabilistic parsing. Studies in Computational Intelligence, 113:229–258.

Nederhof, M.-J. and Satta, G. (2011). Prefix Probability for Probabilistic Synchronous Context-Free Grammars. ACL, pages 460–469.

Nešetřil, J., Milková, E., and Nešetřilová, H. (2001). Otakar Borůvka on minimum spanning tree problem: Translation of both the 1926 papers, comments, history. Discrete Mathematics, 233(1-3):3–36.

Nesson, R., Rush, A., and Shieber, S. M. (2006). Induction of probabilistic synchronous tree-insertion grammars for machine translation. In Proceedings of the 7th Conference of the Association for Machine Translation in the Americas, pages 128–137.

Nießen, S., Och, F. J., Leusch, G., and Ney, H. (2000). An evaluation tool for machine translation: fast evaluation for MT research. In Proceedings of the 2nd International Conference on Language Resources and Evaluation.

Nivre, J. (2003). An efficient algorithm for projective dependency parsing. In Proceedings of the 8th International Workshop on Parsing Technologies, pages 149–160.

Nivre, J., Hall, J., and Nilsson, J. (2006). MaltParser: A Data-Driven Parser-Generator for Dependency Parsing. In Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC-2006), pages 2216–2219.

Och, F. J. (2003). Minimum error rate training in statistical machine translation. In Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, volume 1, pages 160–167.

Och, F. J. and Ney, H. (2003). A Systematic Comparison of Various Statistical Alignment Models. Computational Linguistics, 29(1):19–51.

Owczarzak, K., van Genabith, J., and Way, A. (2007a). Dependency-based automatic evaluation for machine translation. In Proceedings of the NAACL-HLT 2007/AMTA Workshop on Syntax and Structure in Statistical Translation, pages 80–87.

Owczarzak, K., van Genabith, J., and Way, A. (2007b). Labelled dependencies in machine translation evaluation. In Proceedings of the Second Workshop on Statistical Machine Translation, pages 104–111.

Padó, S., Galley, M., Jurafsky, D., and Manning, C. D. (2009). Robust Machine Translation Evaluation with Entailment Features. In Proceedings of the 47th Annual Meeting of the ACL and the 4th IJCNLP of the AFNLP, pages 297–305, Suntec, Singapore. Association for Computational Linguistics.

Paolacci, G., Chandler, J., and Ipeirotis, P. G. (2010). Running experiments on Amazon Mechanical Turk. Judgment and Decision making, 5(5):411–419.

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 of the Association for Computational Linguistics, pages 311–318.

Pawlik, M. and Augsten, N. (2011). RTED: A Robust Algorithm for the Tree Edit Distance. In Proceedings of the 38th International Conference on Very Large Data Bases, pages 334–345.

Petrov, S., Barrett, L., Thibaux, R., and Klein, D. (2006). Learning accurate, compact, and interpretable tree annotation. In Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics, pages 433–440. Association for Computational Linguistics.

Petrov, S. and Klein, D. (2007). Improved inference for unlexicalized parsing. Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference.

Popović, M. (2011). Hjerson: An open source tool for automatic error classification of machine translation output. The Prague Bulletin of Mathematical Linguistics, 96(October):59–68.

Popović, M. (2012a). Class error rates for evaluation of machine translation output. In Proceedings of the Seventh Workshop on Statistical Machine Translation, pages 71–75. Association for Computational Linguistics.

Popović, M. (2012b). rgbF: An Open Source Tool for n-gram Based Automatic Evaluation of Machine Translation Output. The Prague Bulletin of Mathematical Linguistics, 98:99–108.

Popović, M. (2015). CHR F: character n-gram F-score for automatic MT evaluation. In Proceedings of the Tenth Workshop on Statistical Machine Translation, pages 392–395, Lisboa, Portugal.

Popović, M., Arčan, M., Avramidis, E., Burchardt, A., and Lommel, A. R. (2015). Poor man's lemmatisation for automatic error classification. In Proceedings of the Eighteenth Annual Conference of the European Association for Machine Translation, pages 105–112.

Popović, M., Avramidis, E., Burchardt, A., Hunsicker, S., Schmeier, S., Tscherwinka, C., Vilar, D., and Uszkoreit, H. (2013). Learning from human judgments of machine translation output. In Proceedings of the MT Summit XIV, pages 231–238, Nice, France.

Popović, M. and Burchardt, A. (2011). From Human to Automatic Error Classification for Machine Translation Output. In Proceedings of the 15th International Conference of the European Association for Machine Translation, pages 265–272, Leuven, Belgium.

Popović, M., de Gispert, A., Gupta, D., Lambert, P., Ney, H., Mariño, J. B., Federico, M., and Banchs, R. (2006). Morpho-syntactic Information for Automatic Error Analysis of Statistical Machine Translation Output. In Proceedings of the Workshop on Statistical Machine Translation, pages 1–6.

Popović, M., Lommel, A. R., Burchardt, A., Avramidis, E., and Uszkoreit, H. (2014). Relations between different types of post-editing operations, cognitive effort and temporal effort. In Proceedings of the 17th Annual Conference of the European Association for Machine Translation, pages 191–198.

Popović, M. and Ney, H. (2007). Word Error Rates: Decomposition over PoS Classes and Applications for Error Analysis. In Proceedings of the Second Workshop on Statistical Machine Translation, pages 48–55, Prague, Czech Republic. Association for Computational Linguistics.

Popović, M. and Ney, H. (2009). Syntax-oriented evaluation measures for machine translation output. In Proceedings of the Fourth Workshop on Statistical Machine Translation, pages 29–32, Athens, Greece.

Popović, M. and Ney, H. (2011). Towards Automatic Error Analysis of Machine Translation Output. Computational Linguistics, 37(4):657–688.

Quinn, A. J. and Bederson, B. B. (2011). Human computation. In Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11, page 1403.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2):257–286.

Rambow, O. (2010). The simple truth about dependency and phrase structure representations: An opinion piece. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pages 337–340. Association for Computational Linguistics.

Ratnaparkhi, A. (1996). A Maximum Entropy Model for Part-of-Speech Tagging. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, volume 1, pages 133–142.

Roark, B. (2001). Probabilistic Top-Down Parsing and Language Modeling. Computational Linguistics, 27(2):249–276.

Roche, E. and Schabes, Y. (1997). Finite-state Language Processing. MIT press.

Sakaguchi, K., Post, M., and Van Durme, B. (2014). Efficient Elicitation of Annotations for Human Evaluation of Machine Translation. In Proceedings of the Ninth Workshop on Statistical Machine Translation, pages 1–11.

Santorini, B. (1990). Part-of-Speech Tagging Guidelines for the Penn Treebank Project (3rd Revision). Technical Report MS-CIS-90-47, University of Pennsylvania.

Schabes, Y. (1991). Polynomial time and space shift-reduce parsing of arbitrary context-free grammars. In Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics, pages 106–113.

Scott, B. and Barreiro, A. (2009). OpenLogos MT and the SAL Representation Language. In Proceedings of the First International Workshop on Free/Open-Source Rule-Based Machine Translation, pages 19–26.

Secară, A. (2005). Translation evaluation - a state of the art survey. In Proceedings of the eCoLoRe/MeLLANGE Workshop, pages 39–44, Leeds.

Sennrich, R. and Haddow, B. (2016). Linguistic Input Features Improve Neural Machine Translation. In Proceedings of the First Conference on Machine Translation, volume 1, pages 83–91.

Shieber, S. M. (2007). Probabilistic synchronous tree-adjoining grammars for machine translation: The argument from bilingual dictionaries. In Proceedings of the NAACL-HLT 2007/AMTA Workshop on Syntax and Structure in Statistical Translation, pages 88–95.

Sipser, M. (1996). Introduction to the Theory of Computation. Course Technology.

Smith, K. S., Specia, L., and Steele, D. (2015). Sheffield Systems for the Finnish-English WMT Translation Task. In Proceedings of the 10th Workshop on Statistical Machine Translation, pages 172–176, Lisboa, Portugal.

Snover, M., Dorr, B. J., Schwartz, R., Micciulla, L., and Makhoul, J. (2006). A Study of Translation Edit Rate with Targeted Human Annotation. In Proceedings of the 7th Conference of the Association for Machine Translation in the Americas, pages 223–231, Cambridge, Massachusetts.

Snover, M., Madnani, N., Dorr, B. J., and Schwartz, R. (2009). Fluency, adequacy, or HTER? Exploring Different Human Judgments with a Tunable MT Metric. In Proceedings of the Fourth Workshop on Statistical Machine Translation, pages 259–268.

Somers, H. (1999). Review article: Example-based machine translation. Machine Translation, 14(2):113–157.

Souter, C. and Atwell, E. (1994). Using Parsed Corpora: A review of current practice. Corpus-Based Research into Language, pages 143–158.

Spearman, C. (1904). The Proof and Measurement of Association between Two Things. The American Journal of Psychology, 15(1):72–101.

Specia, L., Raj, D., and Turchi, M. (2010). Machine translation evaluation versus quality estimation. Machine Translation, 24(1):39–50.

Stanojević, M. and Sima'an, K. (2014a). BEER: BEtter Evaluation as Ranking. Proceedings of the Ninth Workshop on Statistical Machine Translation, pages 414–419.

Stanojević, M. and Sima'an, K. (2014b). Evaluating Word Order Recursively over Permutation-Forests. In Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, pages 138–147.

Steedman, M. (2000). The syntactic process, volume 35. MIT Press.

Steinberger, R., Pouliquen, B., Widiger, A., Ignat, C., Erjavec, T., Tufis, D., and Varga, D. (2006). The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In Proceedings of the 5th International Conference on Language Resources and Evaluation.

Stine, R. (1989). An Introduction to Bootstrap Methods. Sociological Methods and Research, 18(2-3):243–291.

Student (1908). The Probable Error of a Mean. Biometrika, 6(1):1–25.

Sumita, E. and Iida, H. (1991). Experiments and prospects of example-based machine translation. In Proceedings of the 29th Annual Meeting on Association for Computational Linguistics, pages 185–192.

Tai, K.-C. (1979). The Tree-to-Tree Correction Problem. Journal of the ACM, 26(3):422–433.

Talbot, D., Kazawa, H., Ichikawa, H., Katz-Brown, J., Seno, M., and Och, F. J. (2011). A lightweight evaluation framework for machine translation reordering. In Proceedings of the Sixth Workshop on Statistical Machine Translation, pages 12–21. Association for Computational Linguistics.

The Apache Software Foundation (2011). Apache OpenNLP. https://opennlp.apache.org/. Last accessed on 5 February 2017.

Tillmann, C., Vogel, S., Ney, H., Zubiaga, A., and Sawaf, H. (1997). Accelerated DP Based Search for Statistical Translation. In Proceedings of the Fifth European Conference on Speech Communication and Technology, pages 2667–2670.

Ting, K. M. (2010). Precision and Recall. In Sammut, C. and Webb, G. I., editors, Encyclopedia of Machine Learning, page 781. Springer US, Boston, MA.

Tomita, M. (1991). Generalized LR parsing. Springer.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In Proceedings of HLT-NAACL 2003, pages 252–259.

Tsujii, J. (1986). Future directions of machine translation. In Proceedings of the Conference on Computational Linguistics, pages 655–668.

Ulam, S. (1972). Some ideas and prospects in biomathematics. Annual Review of Biophysics and Bioengineering, 1(1):277–292.

University of St Andrews (2015). Reports and Financial Statements of the University Court for the year to 31 July 2015. Technical Report July, University of St Andrews.

van Rijsbergen, C. J. (1979). Information Retrieval. Butterworth-Heinemann, Newton, MA, USA, 2nd edition.

Vijay-Shanker, K., Weir, D. J., and Joshi, A. K. (1988). Characterizing structural descriptions produced by various grammatical formalisms. In Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics, pages 104–111.

Vilar, D., Popović, M., and Ney, H. (2006a). AER: Do we need to "improve" our alignments? In Proceedings of the International Workshop on Spoken Language Translation, pages 205–212.

Vilar, D., Xu, J., D'Haro, L. F., and Ney, H. (2006b). Error analysis of statistical machine translation output. In Proceedings of the Conference on Language Resources and Evaluation, pages 697–702, Genoa.

Vogel, S., Ney, H., and Tillmann, C. (1996). HMM-based Word Alignment in Statistical Machine Translation. In Proceedings of the 16th conference on Computational Linguistics, pages 836–841.

Voutilainen, A. (2003). Part-of-speech tagging. In The Oxford Handbook of Computational Linguistics, chapter 11, pages 219–232. Oxford University Press.

Wagner, R. A. and Fischer, M. J. (1974). The String-to-String Correction Problem. Journal of the ACM, 21(1):168–173.

Wang, M. and Manning, C. D. (2012). SPEDE: probabilistic edit distance metrics for MT evaluation. In Proceedings of the Seventh Workshop on Statistical Machine Translation, pages 76–83.

Weaver, W. (1955). Translation. Machine Translation of Languages, 14:15–23.

Weischedel, R., Palmer, M., and Marcus, M. P. (2011). OntoNotes Release 4.0. Technical report, BBN Technologies.

Yamada, K. and Knight, K. (2001). A syntax-based statistical translation model. In Proceedings of the 39th Annual Meeting on Association for Computational Linguistics, pages 523–530. Association for Computational Linguistics.

Yang, M. and Kirchhoff, K. (2006). Phrase-Based Backoff Models for Machine Translation of Highly Inflected Languages. In Proceedings of the 21st International Conference on Computational Linguistics, pages 41–48.

Younger, D. H. (1967). Recognition and parsing of context-free languages in time n^3. Information and Control, 10(2):189–208.

Yu, H., Wu, X., Xie, J., Jiang, W., Liu, Q., and Lin, S. (2014). RED: A Reference Dependency Based MT Evaluation Metric. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 2042–2051.

Zalmout, N. and Habash, N. (2017). Optimizing Tokenization Choice for Machine Translation across Multiple Target Languages. The Prague Bulletin of Mathematical Linguistics, 108(June):257–269.

Zeman, D., Fishel, M., Berka, J., and Bojar, O. (2011). Addicter: What is Wrong with My Translations? The Prague Bulletin of Mathematical Linguistics, 96:79–88.

Zens, R. and Bender, O. (2005). The RWTH Phrase-based Statistical Machine Translation System. In Proceedings of the International Workshop on Spoken Language Translation, pages 145–152.

Appendix A

WOJ-DB

Pilot study

Pages 160 and 161 contain demographic information about participants to the pilot study we ran for WOJ-DB (Section 6.4). Figure A.1 contains information about score distributions and ranges across all five questionnaires, similarly to Figure 6.1 (page 96) for the main survey. Figure A.2 presents the lengths of the sentences included in the pilot study. As with the main survey (Figure 6.2, page 97) participants were rarely shown sentence pairs for which the hypothesis and reference differed dramatically in length. Figure A.3 summarises information about our pilot participants' education levels and interest in grammar, just as Figure 6.6 does for the main study. Finally, Figure A.4 indicates the languages spoken by pilot study participants, presented in the same manner as Figure 6.7 (page 104).

Participant information sheets

Pages 162 to 166 reproduce information given to participants of WOJ-DB during their participation. The information sheet, provided on arrival and providing limited background information on the survey, is on pages 162-163. Pages 164-165 show the consent form each participant was required to sign before participating. After completing the survey, participants were provided with the debriefing sheet shown on page 166. Page 167 shows the first page of the survey, requesting personal information about the participant as discussed in Section 6.3.2 and providing them with a sample sentence as per Section 6.3.2.

Shared sentences

While questionnaires in WOJ-DB contained mostly unique sentence pairs, 20 questions out of 50 were the same for all participants, as described in Section 6.3.4. Half of these sentences were real hypothesis/reference pairs produced for WMT, while the other half were automatic permutations of those reference sentences. The WMT hypotheses and references, along with brief descriptors of the important factors leading to their selection, are shown in the tables on pages 168 and 169.

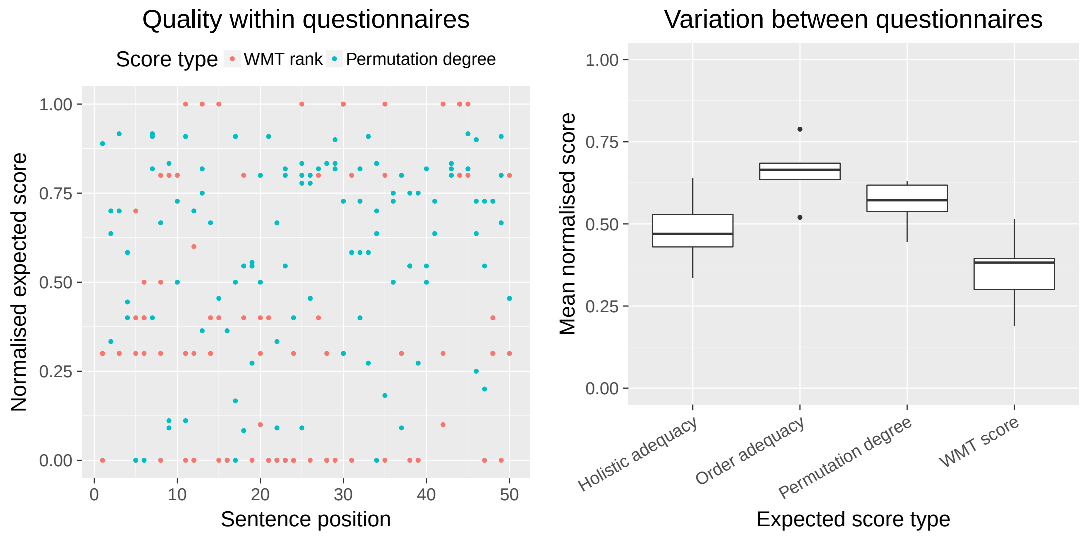


Figure A.1: Quality distribution between pilot study questionnaires, based on sentence type, WMT rank, permutation degree or survey scores; and within questionnaires by position and sentence type

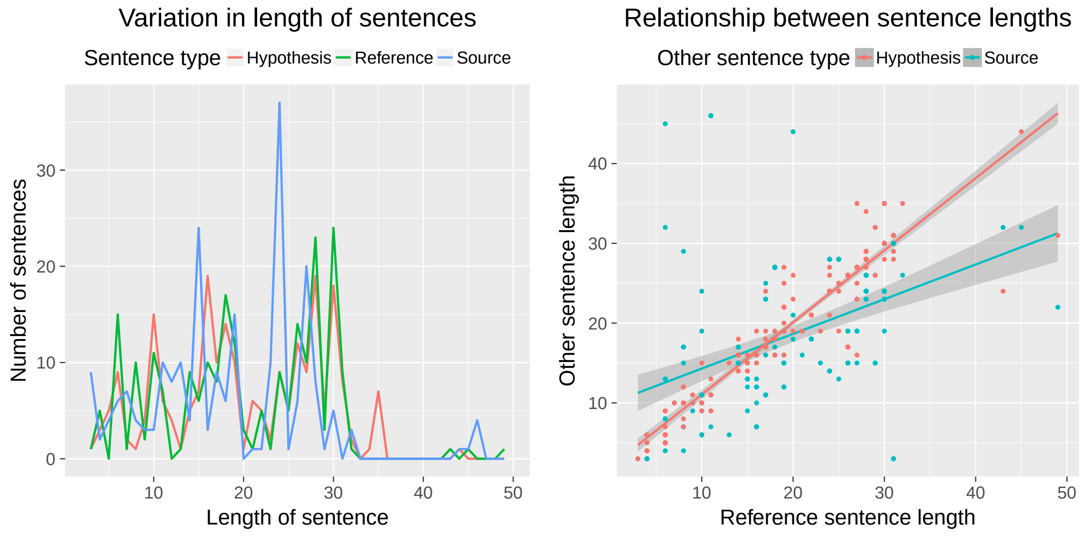


Figure A.2: Sentence length variation within all sentences used in the WOJ-DB pilot study

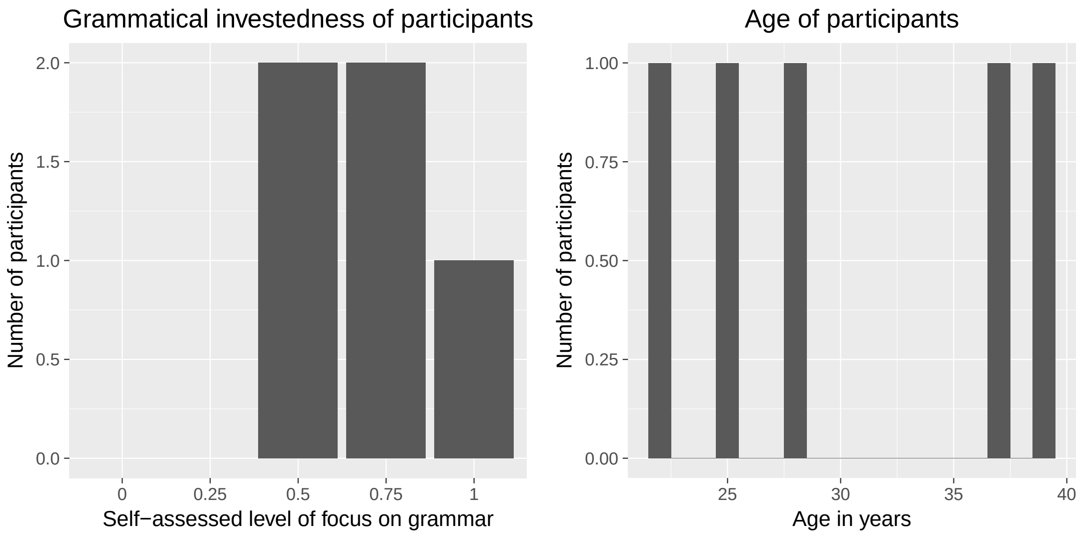


Figure A.3: Education level and grammatical focus of participants

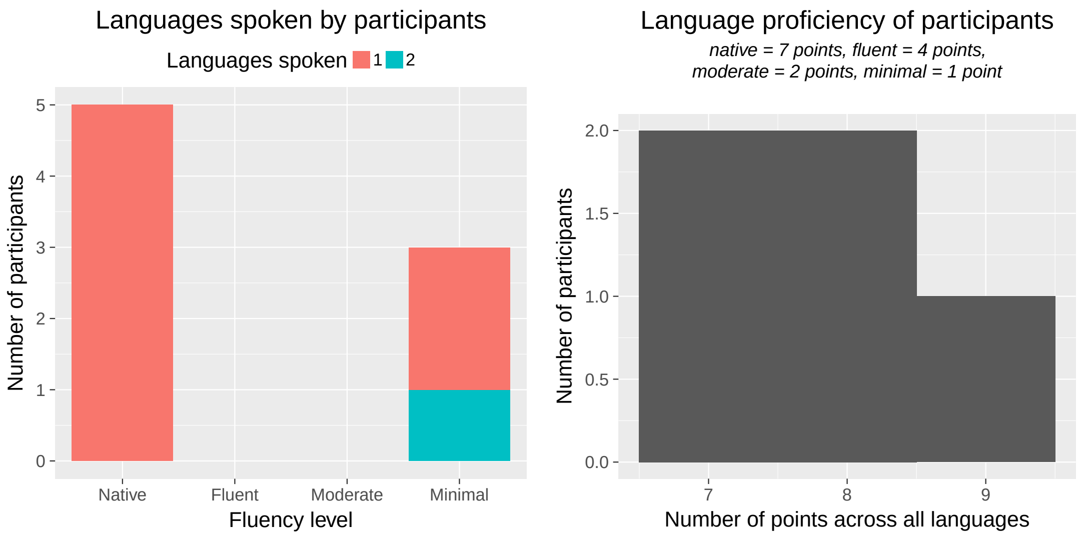
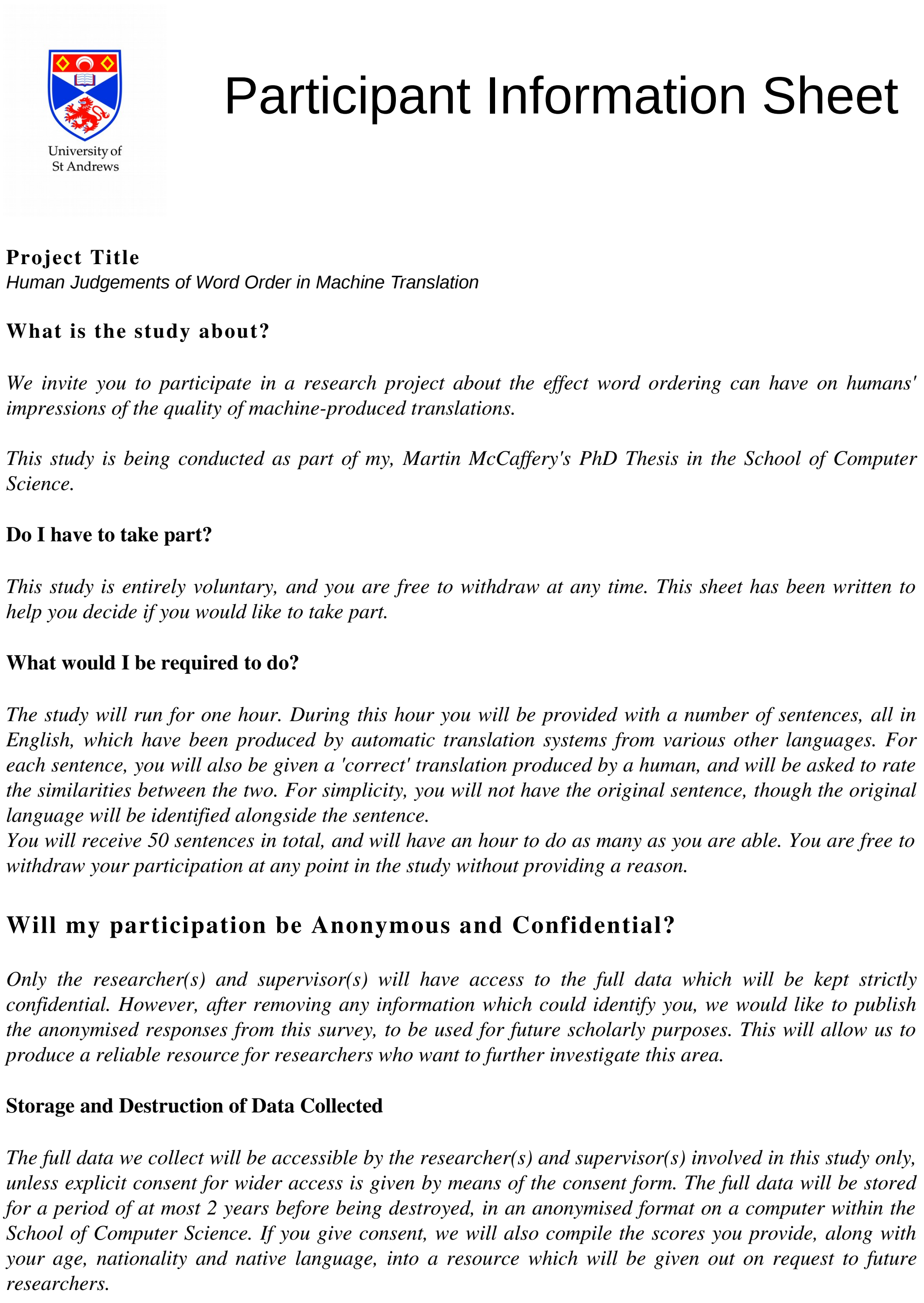


Figure A.4: Language knowledge of participants in the WOJ-DB pilot study



University of St Andrews

Participant Information Sheet

Project Title

Human Judgements of Word Order in Machine Translation

What is the study about?

We invite you to participate in a research project about the effect word ordering can have on humans' impressions of the quality of machine-produced translations.

This study is being conducted as part of my, Martin McCaffery's PhD Thesis in the School of Computer Science.

Do I have to take part?

This study is entirely voluntary, and you are free to withdraw at any time. This sheet has been written to help you decide if you would like to take part.

What would I be required to do?

The study will run for one hour. During this hour you will be provided with a number of sentences, all in English, which have been produced by automatic translation systems from various other languages. For each sentence, you will also be given a 'correct' translation produced by a human, and will be asked to rate the similarities between the two. For simplicity, you will not have the original sentence, though the original language will be identified alongside the sentence.
You will receive 50 sentences in total, and will have an hour to do as many as you are able. You are free to withdraw your participation at any point in the study without providing a reason.

Will my participation be Anonymous and Confidential?

Only the researcher(s) and supervisor(s) will have access to the full data which will be kept strictly confidential. However, after removing any information which could identify you, we would like to publish the anonymised responses from this survey, to be used for future scholarly purposes. This will allow us to produce a reliable resource for researchers who want to further investigate this area.

Storage and Destruction of Data Collected

The full data we collect will be accessible by the researcher(s) and supervisor(s) involved in this study only, unless explicit consent for wider access is given by means of the consent form. The full data will be stored for a period of at most 2 years before being destroyed, in an anonymised format on a computer within the School of Computer Science. If you give consent, we will also compile the scores you provide, along with your age, nationality and native language, into a resource which will be given out on request to future researchers.

What will happen to the results of the research study?

The ratings you provide will be statistically compared with automatically-generated ratings on word order quality. This will be used to assess the quality of the automatic judgements, with the human ones assumed to be correct.

The results will be finalised during 2017 and written up as part of the researcher's PhD Thesis. They may also be published in paper format.

Reward

You will be given a £5 Amazon voucher for your participation in this survey.

Are there any potential risks to taking part?

There are no known risks associated with participation in this study.

Questions

You will have the opportunity to ask any questions in relation to this project before completing a Consent Form.

Consent and Approval

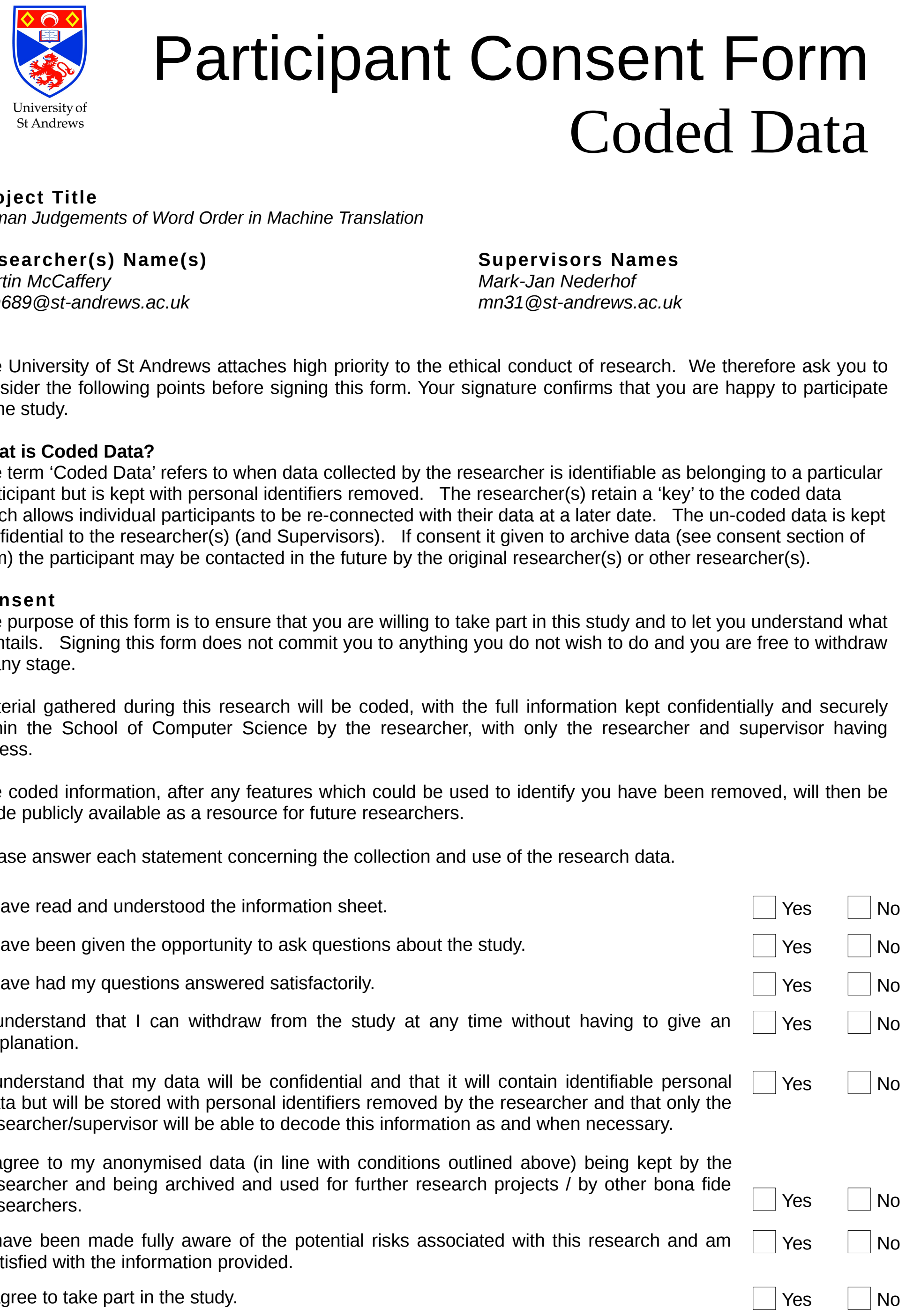
This research proposal has been scrutinised and been granted Ethical Approval through the University ethical approval process.

What should I do if I have concerns about this study?

A full outline of the procedures governed by the University Teaching and Research Ethical Committee is available at http://www.st-andrews.ac.uk/utrec/guidelinespolicies/complaints/

Contact Details

Researcher:	*Martin McCaffery*
Contact Details:	*mm689@st-andrews.ac.uk*
Supervisor:	*Mark-Jan Nederhof*
Contact Details:	*mn31@st-andrews.ac.uk*



University of St Andrews

Participant Consent Form
Coded Data

Project Title
Human Judgements of Word Order in Machine Translation

Researcher(s) Name(s)
Martin McCaffery
mm689@st-andrews.ac.uk

Supervisors Names
Mark-Jan Nederhof
mn31@st-andrews.ac.uk

The University of St Andrews attaches high priority to the ethical conduct of research. We therefore ask you to consider the following points before signing this form. Your signature confirms that you are happy to participate in the study.

What is Coded Data?
The term 'Coded Data' refers to when data collected by the researcher is identifiable as belonging to a particular participant but is kept with personal identifiers removed. The researcher(s) retain a 'key' to the coded data which allows individual participants to be re-connected with their data at a later date. The un-coded data is kept confidential to the researcher(s) (and Supervisors). If consent it given to archive data (see consent section of form) the participant may be contacted in the future by the original researcher(s) or other researcher(s).

Consent
The purpose of this form is to ensure that you are willing to take part in this study and to let you understand what it entails. Signing this form does not commit you to anything you do not wish to do and you are free to withdraw at any stage.

Material gathered during this research will be coded, with the full information kept confidentially and securely within the School of Computer Science by the researcher, with only the researcher and supervisor having access.

The coded information, after any features which could be used to identify you have been removed, will then be made publicly available as a resource for future researchers.

Please answer each statement concerning the collection and use of the research data.

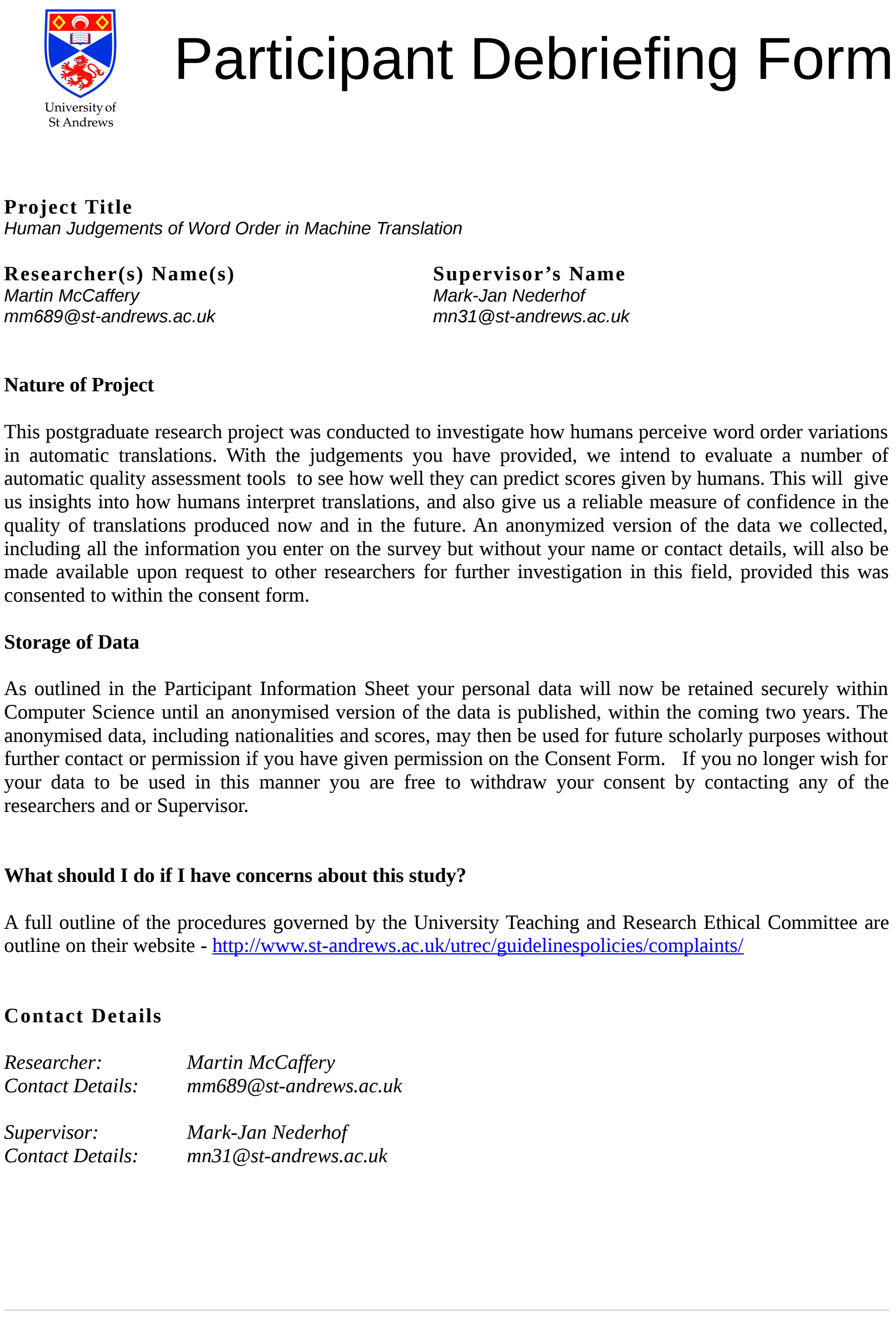
I have read and understood the information sheet.	□ Yes	□ No
I have been given the opportunity to ask questions about the study.	□ Yes	□ No
I have had my questions answered satisfactorily.	□ Yes	□ No
I understand that I can withdraw from the study at any time without having to give an explanation.	□ Yes	□ No
I understand that my data will be confidential and that it will contain identifiable personal data but will be stored with personal identifiers removed by the researcher and that only the researcher/supervisor will be able to decode this information as and when necessary.	□ Yes	□ No
I agree to my anonymised data (in line with conditions outlined above) being kept by the researcher and being archived and used for further research projects / by other bona fide researchers.	□ Yes	□ No
I have been made fully aware of the potential risks associated with this research and am satisfied with the information provided.	□ Yes	□ No
I agree to take part in the study.	□ Yes	□ No

Participation in this research is completely voluntary and your consent is required before you can participate in this research. If you decide at a later date that data should be destroyed we will honour your request in writing.

Name in Block Capitals ______________________________

Signature ______________________________

Date ______________________________



University of
St Andrews

Participant Debriefing Form

Project Title
Human Judgements of Word Order in Machine Translation

Researcher(s) Name(s)	**Supervisor's Name**
Martin McCaffery	*Mark-Jan Nederhof*
mm689@st-andrews.ac.uk	*mn31@st-andrews.ac.uk*

Nature of Project

This postgraduate research project was conducted to investigate how humans perceive word order variations in automatic translations. With the judgements you have provided, we intend to evaluate a number of automatic quality assessment tools to see how well they can predict scores given by humans. This will give us insights into how humans interpret translations, and also give us a reliable measure of confidence in the quality of translations produced now and in the future. An anonymized version of the data we collected, including all the information you enter on the survey but without your name or contact details, will also be made available upon request to other researchers for further investigation in this field, provided this was consented to within the consent form.

Storage of Data

As outlined in the Participant Information Sheet your personal data will now be retained securely within Computer Science until an anonymised version of the data is published, within the coming two years. The anonymised data, including nationalities and scores, may then be used for future scholarly purposes without further contact or permission if you have given permission on the Consent Form.   If you no longer wish for your data to be used in this manner you are free to withdraw your consent by contacting any of the researchers and or Supervisor.

What should I do if I have concerns about this study?

A full outline of the procedures governed by the University Teaching and Research Ethical Committee are outline on their website - http://www.st-andrews.ac.uk/utrec/guidelinespolicies/complaints/

Contact Details

Researcher:	*Martin McCaffery*
Contact Details:	*mm689@st-andrews.ac.uk*
Supervisor:	*Mark-Jan Nederhof*
Contact Details:	*mn31@st-andrews.ac.uk*

Martin McCaffery
Word Ordering in Machine Translation

This questionnaire is automatically read by a computer program. Please use a pen when filling in your answers.

Select an answer: ☒ Cancel an answer, to change it: ■

1 About You

1.1 Is English your first language? If not, how long would you say you have been speaking it fluently?

□ Yes No □

1.2 Please indicate the languages you speak (including English), along with your fluency level and any qualifications you have achieved in each.

□

1.3 What gender best describes you?

□ Male □ Female □ Other □ Prefer not to say

1.4 What nationality do you feel best represents you?

□

1.5 What is the highest level of education you have completed?

□ School or none □ College diploma □ Undergraduate □ Masters degree □ Ph.D. or higher

1.6 Do you have any disabilities which could impact your reading, e.g. dyslexia?

□ No Yes (please specify) □

1.8 How old are you?

□

1.7 To what extent would you say that you notice grammar and sentence structure in everyday life?

It doesn't affect me at all □ □ □ □ □ Correct use of language is important

2 Sample Sentence

Machine-produced translation:
The man saw the dog for doctor, as ill over day.

Human-produced translation:
The man went to see the vet about his dog, which had been ill for days.

2.1 How difficult would it be to grasp the meaning of the second sentence if you were only shown the first?
There are numerous mistakes in the sentence, which could make its meaning hard to grasp. Does the correct meaning, shown in the second sentence, still clearly exist? If not, is it impossible to work it out?

Very difficult □ □ □ □ □ Very easy

2.2 How much does the ordering of the words on its own cloud the meaning of the first sentence?
*The only ordering mistake is swapping 'dog' and 'vet/doctor' - but this mistake could be very confusing. Consider how many words have been moved **and** how significantly this obscures the meaning of the sentence.*

Enormously □ □ □ □ □ Not at all

Sentence ID, Hypothesis, Reference	Details
wmt14_rbmt1.0.fr-en #905 The program will more focus on the “problems of the real world”, in particular financial mathematics. The syllabus will place a greater focus on “real world problems,” including financial mathematics.	**Source:** French **Length:** short **Quality:** high **Order:** slight
wmt16.pjatk.4520.cs-en #1555 No offense, but we don’t know, “admitted supporter Tripoli Goian. No hard feelings, but I’ve never heard of Lafata, admits Tripolis mainstay Goian.	**Source:** Czech **Length:** short **Quality:** medium **Order:** prime
wmt16.online-g.0.de-en #1654 Criticism of the city boss exerts on the distribution of refugees within the country. The town chief is criticising the allocation of refugees within the federal states.	**Source:** German **Length:** short **Quality:** medium **Order:** prime
wmt16.pjatk.4520.cs-en #877 Replace the Czech Latecoere for Brazilian or European manufacturer is either impossible or very difficult. Replacing the Czech branch of Latecoere would be very complicated, if not impossible, for the Brazilian or European producer.	**Source:** Czech **Length:** medium **Quality:** medium **Order:** slight
wmt14_iitb-ranked-ppl.3173.hi-en #392 American health statistics , the average American male in the 21st century , which is 7 inches , 39 . The waist of the average 21st-century American male is 39.7 inches, according to U.S. health statistics.	**Source:** Hindi **Length:** medium **Quality:** medium **Order:** equal
wmt14_iitb-ranked-ppl.3173.hi-en #674 Wednesday , investigators reviewed surveillance video released by the South Georgia authorities all the way . A southern Georgia judge on Wednesday ordered authorities to release all surveillance video that investigators reviewed.	**Source:** Hindi **Length:** medium **Quality:** low **Order:** equal

Table A.1: Broad descriptions of the sentences shared across every survey (first line in table 6.3), continued in Table A.2.
Sentence IDs indicate the WMT year, translation system and sentence pair number within the corpus translated by that system. The ‘Order’ column indicates the relevance of the ordering of the words in the hypothesis sentence to any confusion of its meaning, relative to other factors such as word choice or omission.

Sentence ID, Hypothesis, Reference	**Details**
wmt14_uedin-wmt14.3422.hi-en #399 The hope to those days is difficult to remember - to remember the one away as they are, have lost a bitter memory of the opportunities. It's hard to remember those days of optimism – they seem a distant memory, a sad reminder of opportunities gone by.	**Source:** Hindi **Length:** medium **Quality:** low **Order:** equal
wmt16.online-f.0.de-en #218 With all building projects in the urban district Cologne is always consulted the museum as specialized office for archaeological ground monument conservation. The museum, as the department for archaeological monument conservation, is always consulted when construction projects are carried out in the city district of Cologne.	**Source:** German **Length:** long **Quality:** low **Order:** prime
wmt16.online-f.0.de-en #2506 Malmö, which struck Celtic in a play-off, in order to reach this stage, began with Anpiff with nine national players on the field and was deeply in the centre zone lined up with a five-man defense and two a heavy opponent. Malmo, who beat Celtic in a play-off to reach this stage, lined up with nine full internationals on the field at kick-off and, with a five-man defence and two deep in midfield, were stuffy opponents.	**Source:** German **Length:** long **Quality:** low **Order:** slight
wmt16.jhu-pbmt.4284.ro-en #853 Striker in Swansea, Michu, associated with Celtic in the transfers, suggested that it could withdraw from the contract with Swans. Swansea striker Michu, linked with Celtic during the transfer window, has dropped hints that he could retire when his contract with the Swans is up.	**Source:** Romanian **Length:** long **Quality:** high **Order:** prime

Table A.2: Continuation of Table A.1

Appendix B

Results

Score ranges

On pages 172 and 173, the ranges of scores produced by each variant of DTED and DERP are presented. As discussed in Section 7.2.1 the absolute scores do not affect our conclusions, which through Spearman's ρ are based instead on simple pairwise comparisons within the data, however the absolute scores may be of interest when comparing our tools to others using other methods.

Unadjusted scores

On pages 174 to 177, we present correlations based on scores unadjusted for random effects as per Section 7.2.3. These thus represent Spearman's ρ values between the judgments provided for WOJ-DB, converted to numerical form as described in Section 6.3.5 but otherwise unaltered.

Holistic scores

Pages 178 and 179 contain graphs similar to those in Chapter 7 but relating to human judgments on holistic adequacy rather than word ordering specifically. These are discussed in more detail in Section 7.3.4.

Participant subsets

Similar graphs appear on pages 180 to 185. These graphs show correlations generated on the subsets of WOJ-DB judgments produced by participants belonging to different demographics, as discussed in Section 7.3.5. We present graphs for participants belonging to different age brackets, linguistic experience and education level. The number of people belonging to any given subgroup is indicated on the relevant graphs.

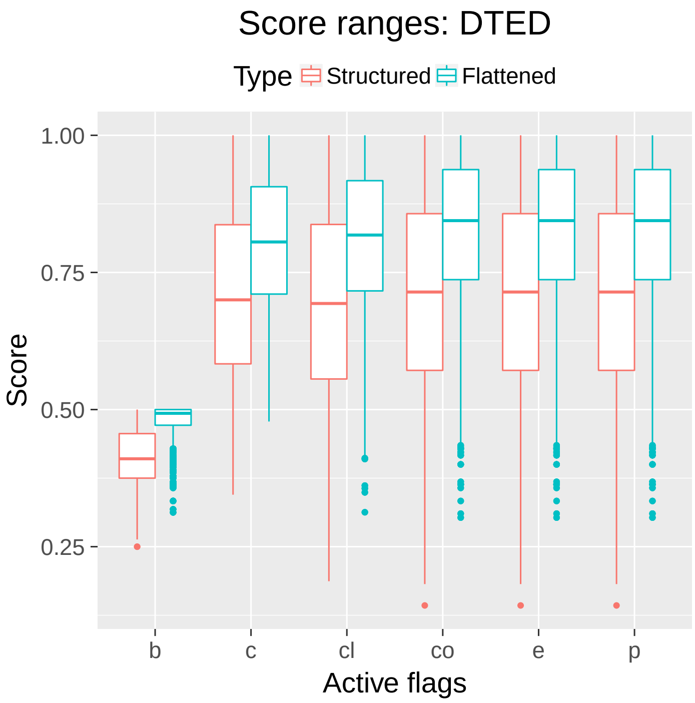


Figure B.1: Score distributions for variants of DTED

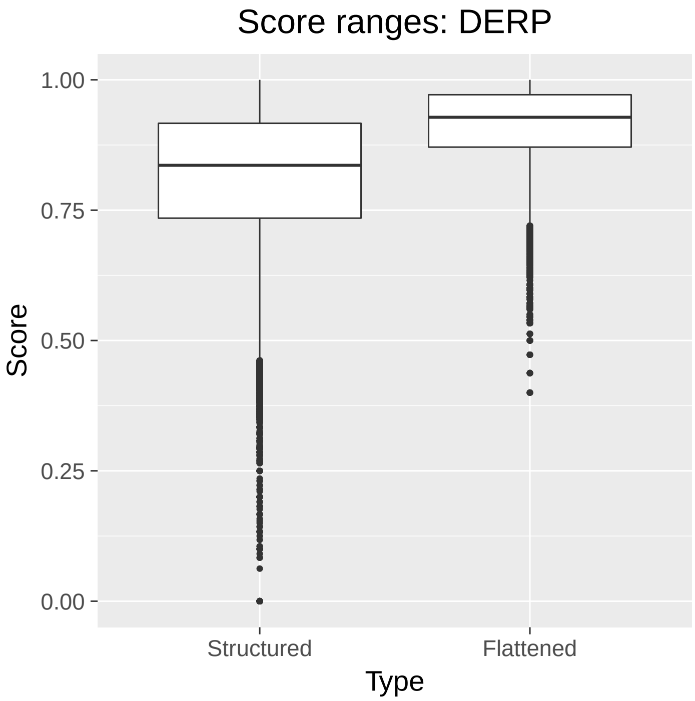


Figure B.2: Score distributions for variants of DERP

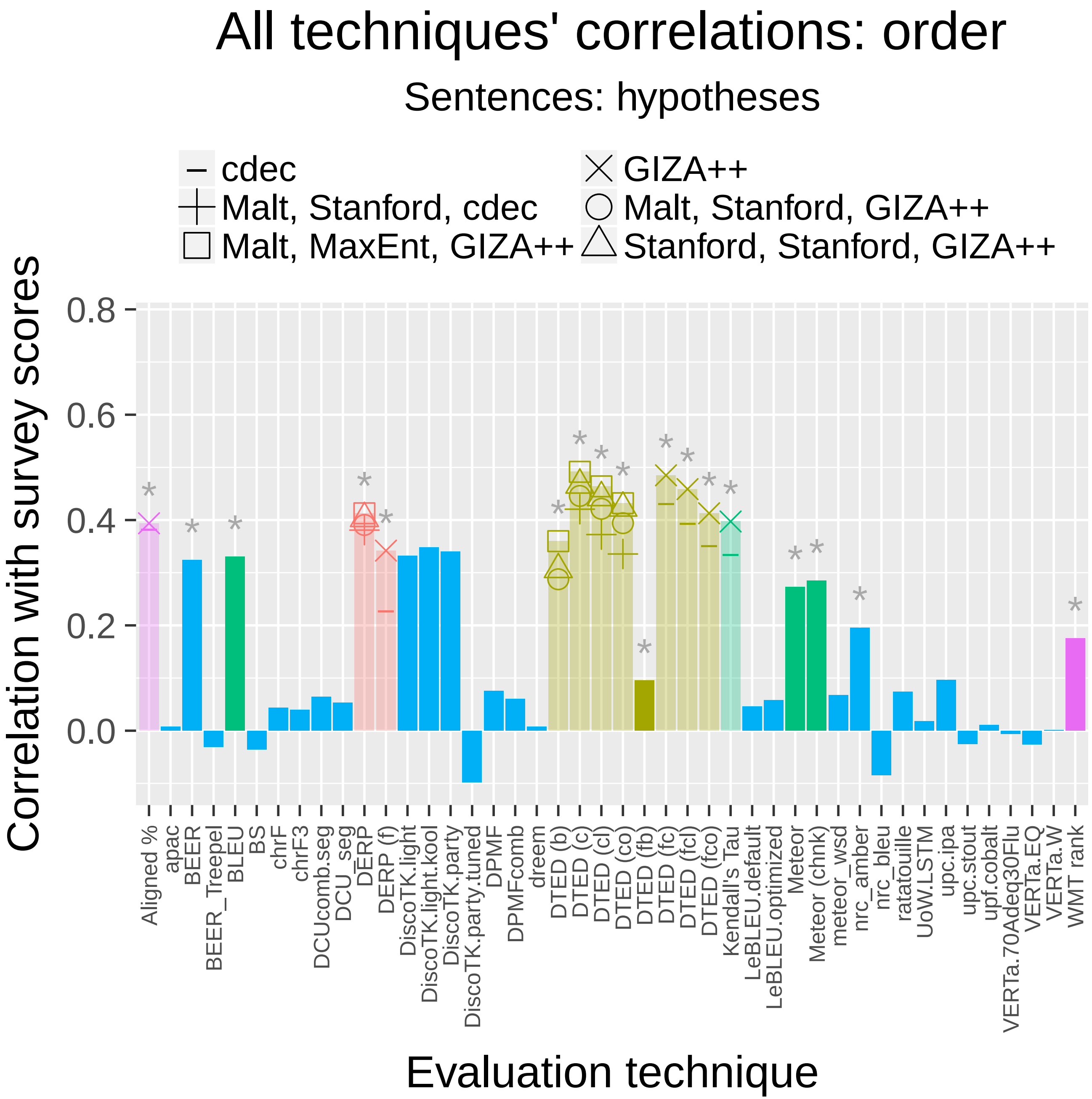


Figure B.3: Correlations between order-focused human judgments and other evaluations across all hypothesis translations in WOJ-DB, without prior adjustment for participants' random effects. Colours are summarised in Table 7.2 (page 117).

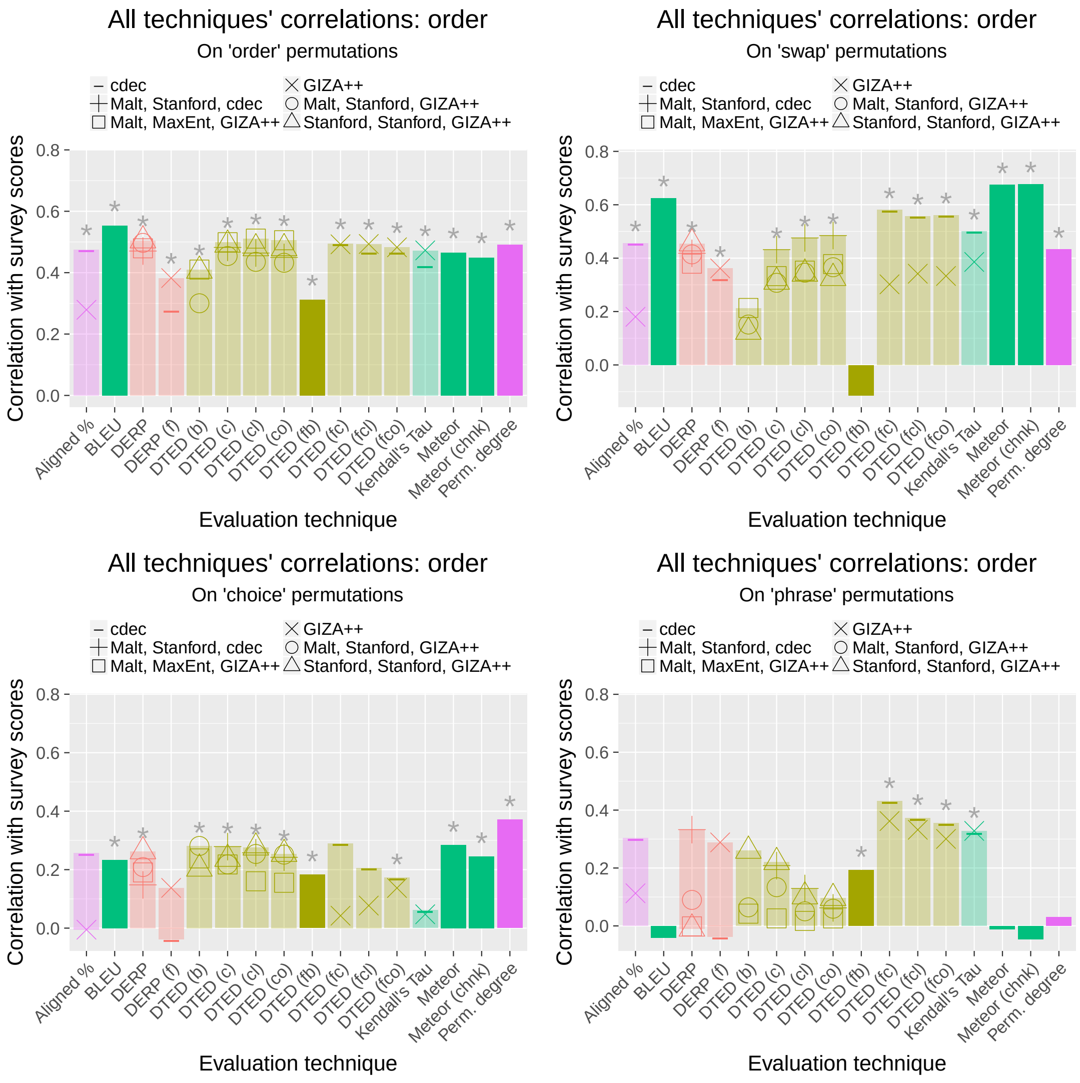


Figure B.4: Correlations between order-focused human judgments and other techniques across each type of automatically permuted sentences in WOJ-DB, without prior adjustment for participants' random effects. Colours are summarised in Table 7.2 (page 117).

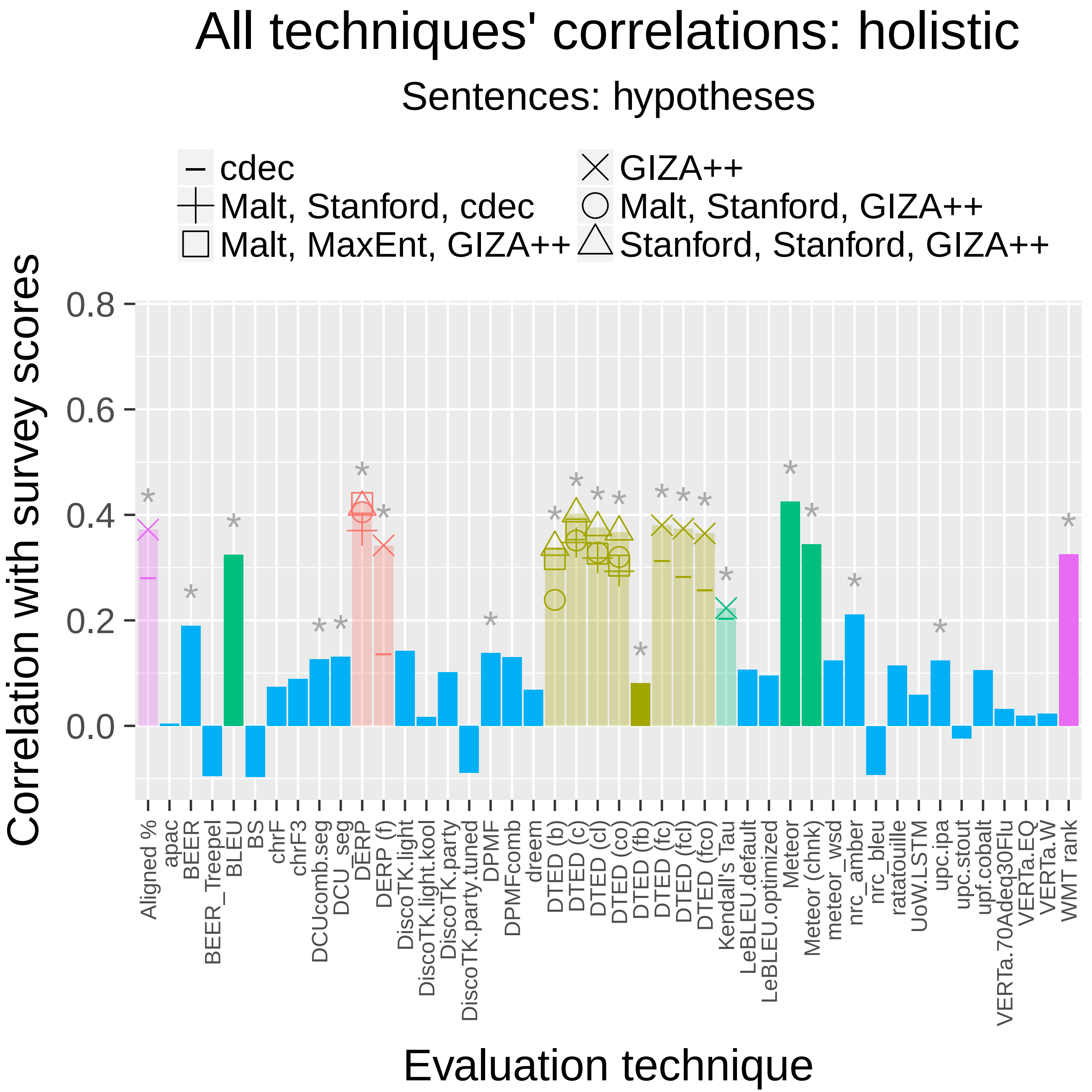


Figure B.5: Correlations between holistic human judgments and other evaluations across all hypothesis translations in WOJ-DB, without prior adjustment for participants' random effects. Colours are summarised in Table 7.2 (page 117).

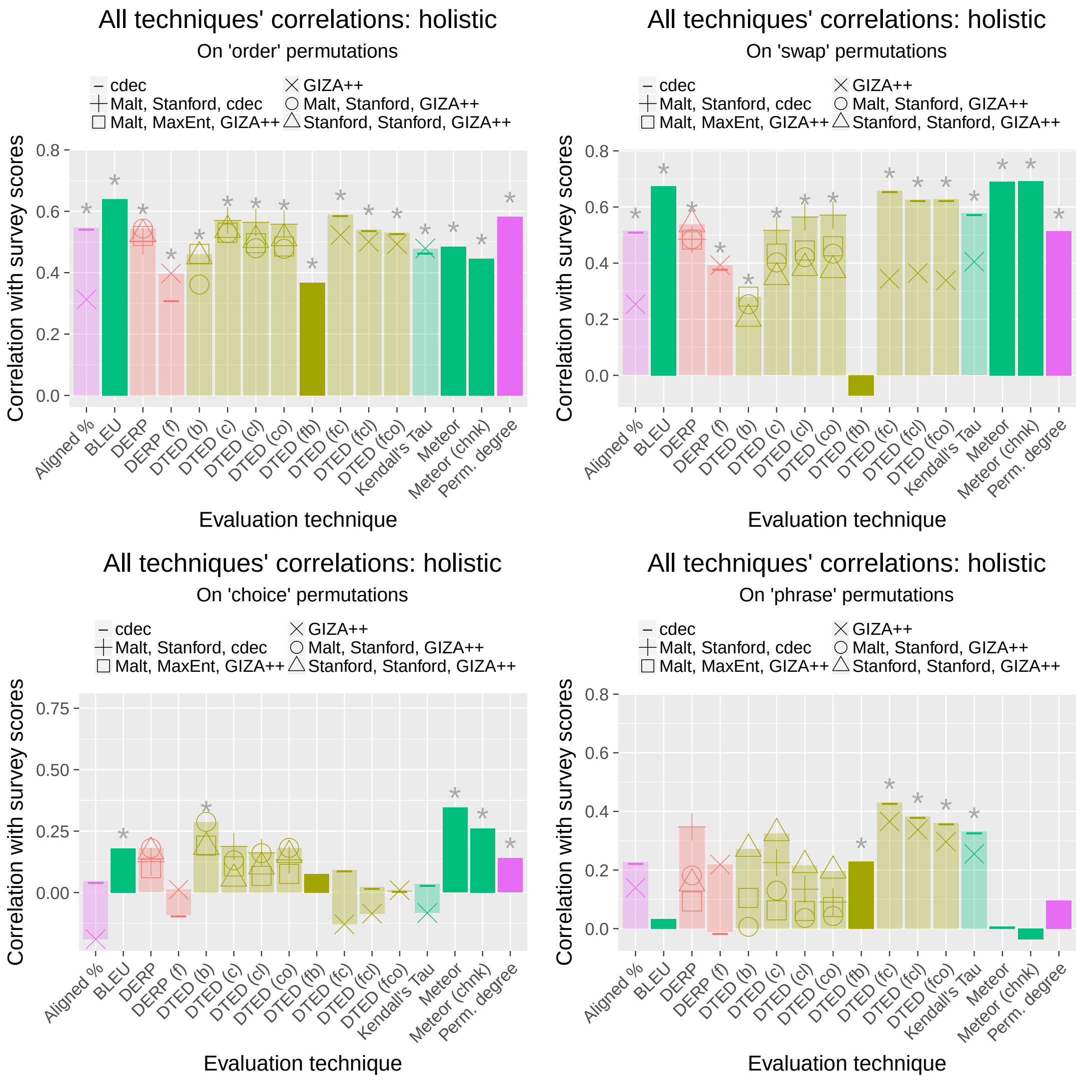


Figure B.6: Correlations between holistic human judgments and other techniques across each type of automatically permuted sentences in WOJ-DB, without prior adjustment for participants' random effects. Colours are summarised in Table 7.2 (page 117).

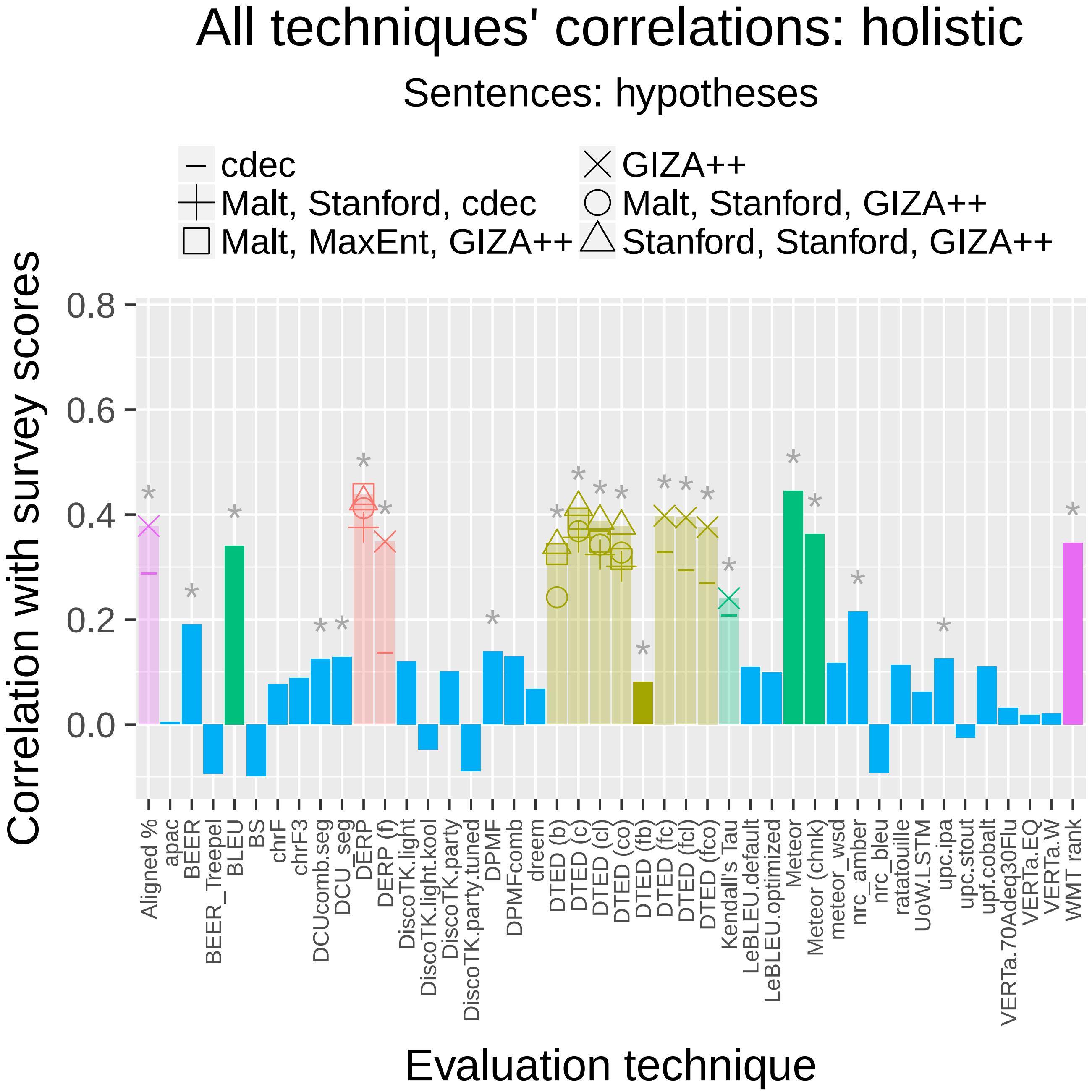


Figure B.7: Correlations between holistic human judgments and other evaluations across all hypothesis translations in WOJ-DB. Colours are summarised in Table 7.2 (page 117).

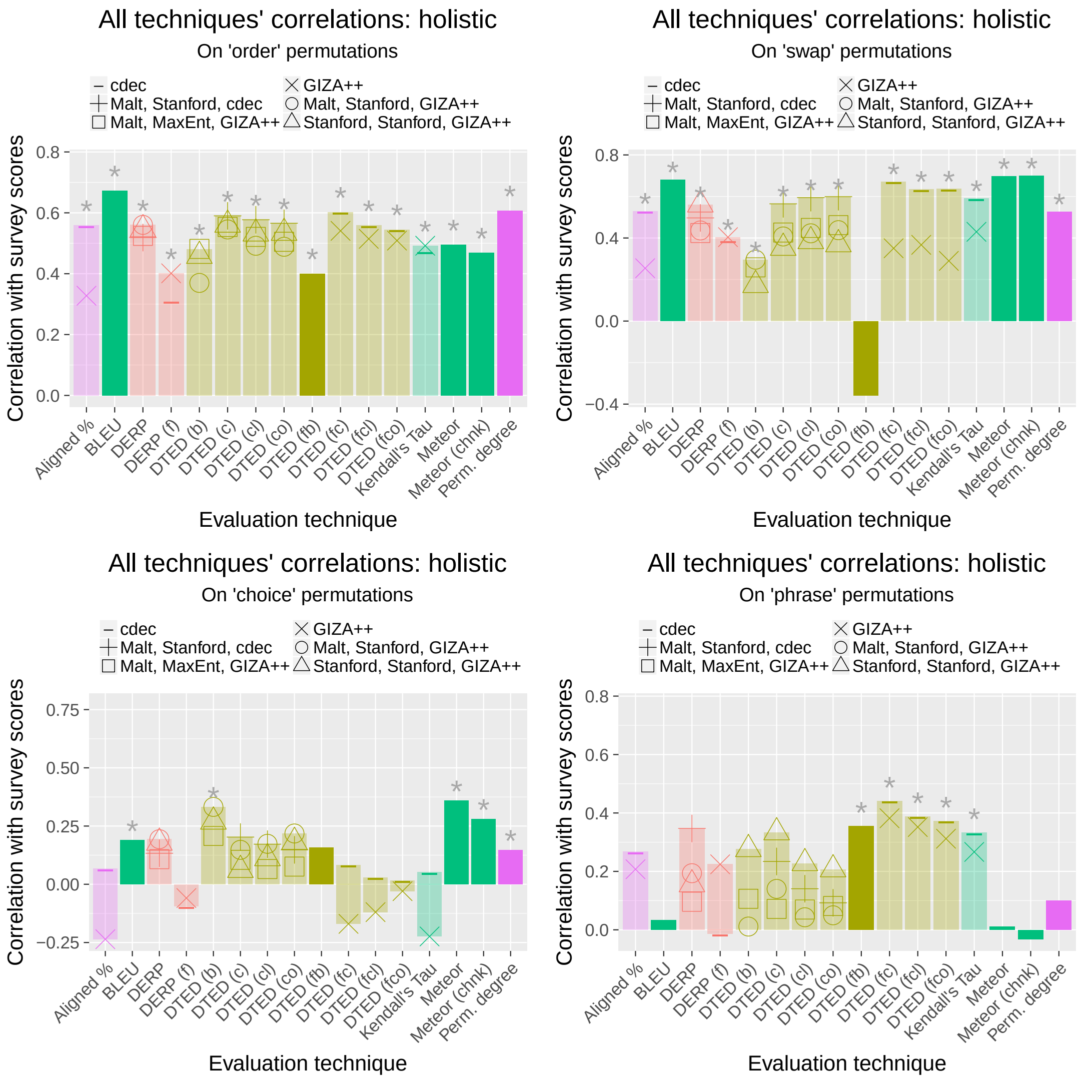


Figure B.8: Correlations between holistic human judgments and other techniques across each type of automatically permuted sentences in WOJ-DB. Colours are summarised in Table 7.2 (page 117).

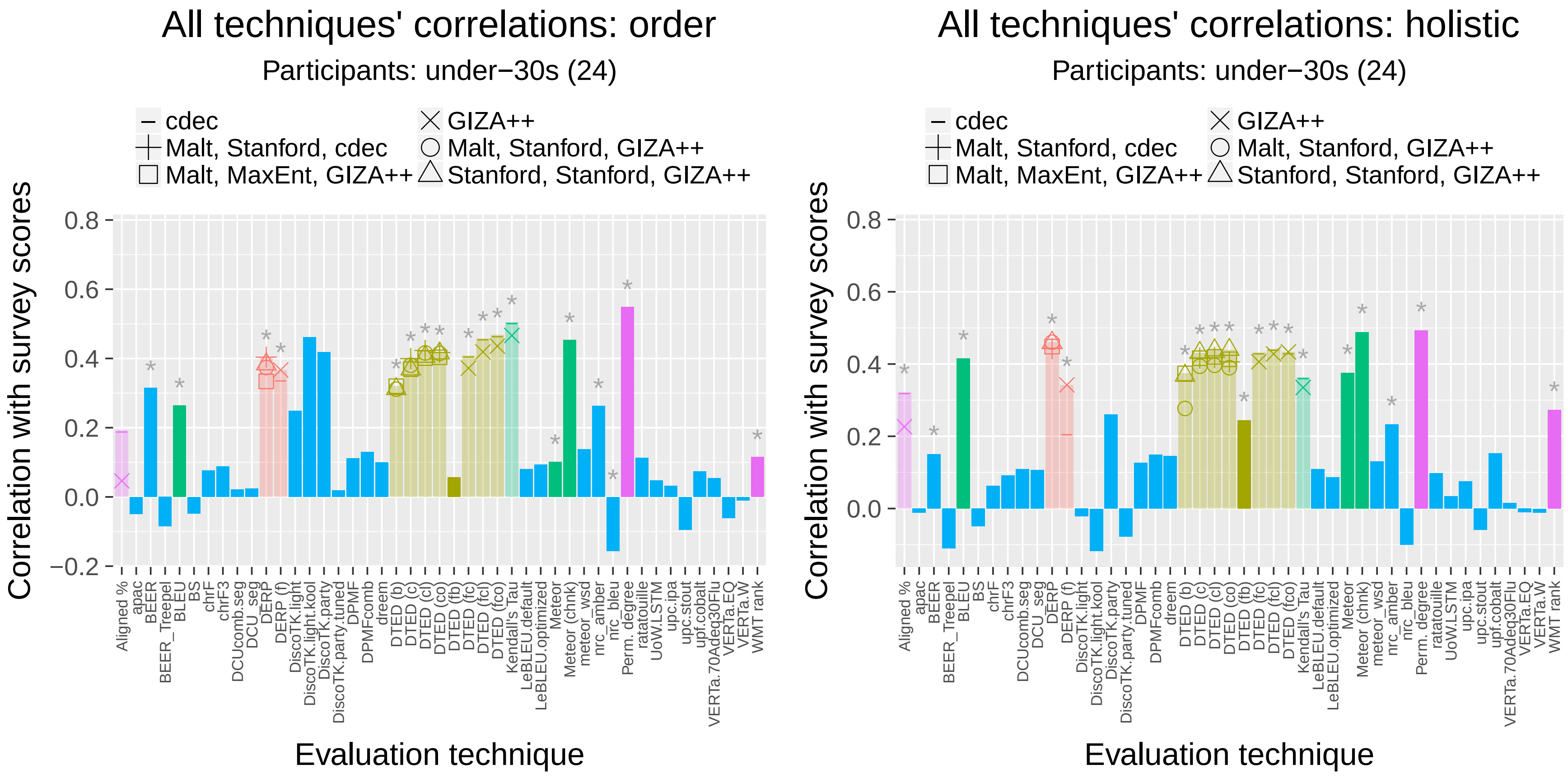


Figure B.9: Correlations between human judgments from WOJ-DB and other techniques for participants aged below 30. Colours are summarised in Table 7.2 (page 117).

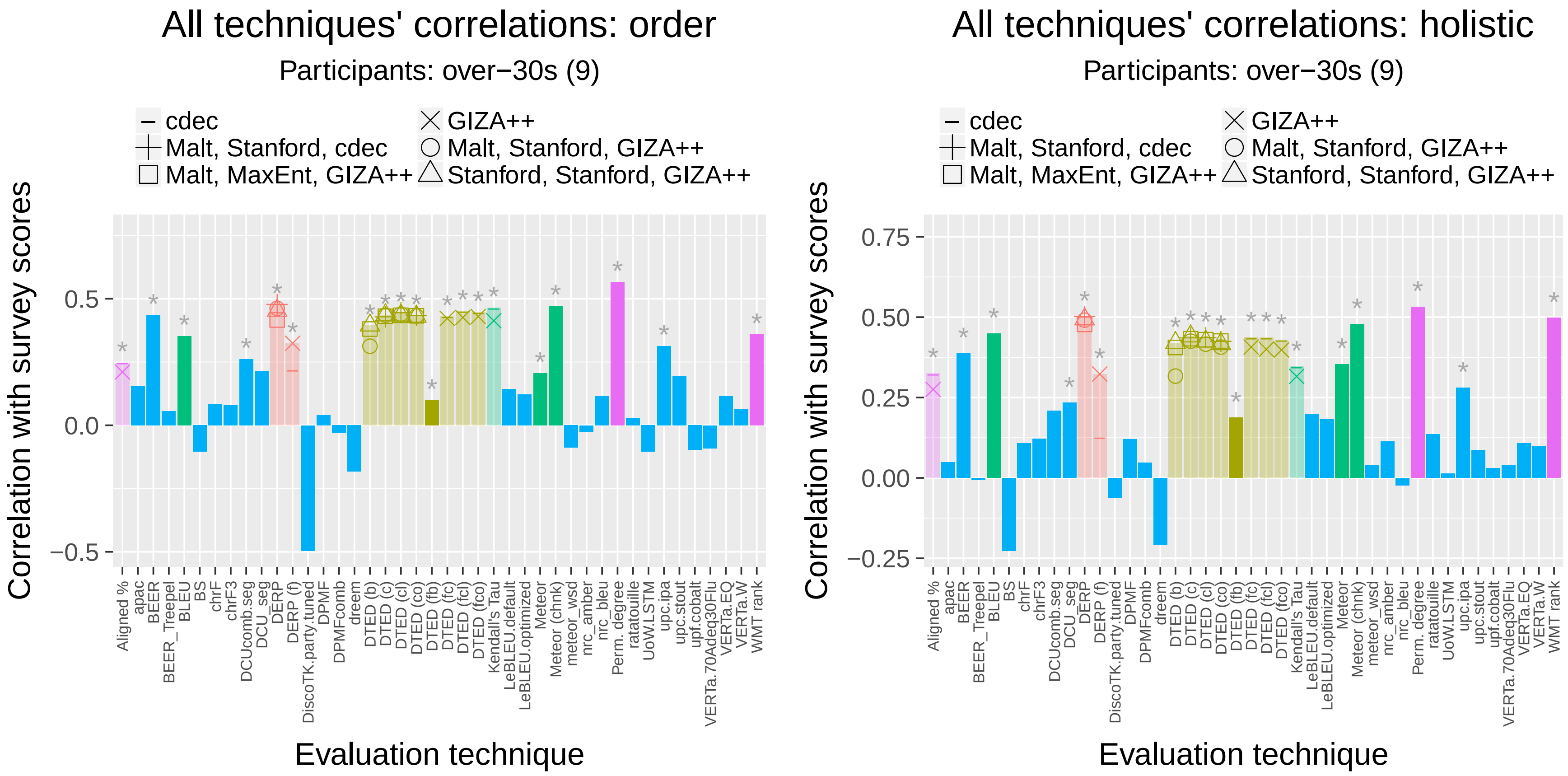


Figure B.10: Correlations between human judgments from WOJ-DB and other techniques for participants aged at least 30. Colours are summarised in Table 7.2 (page 117).

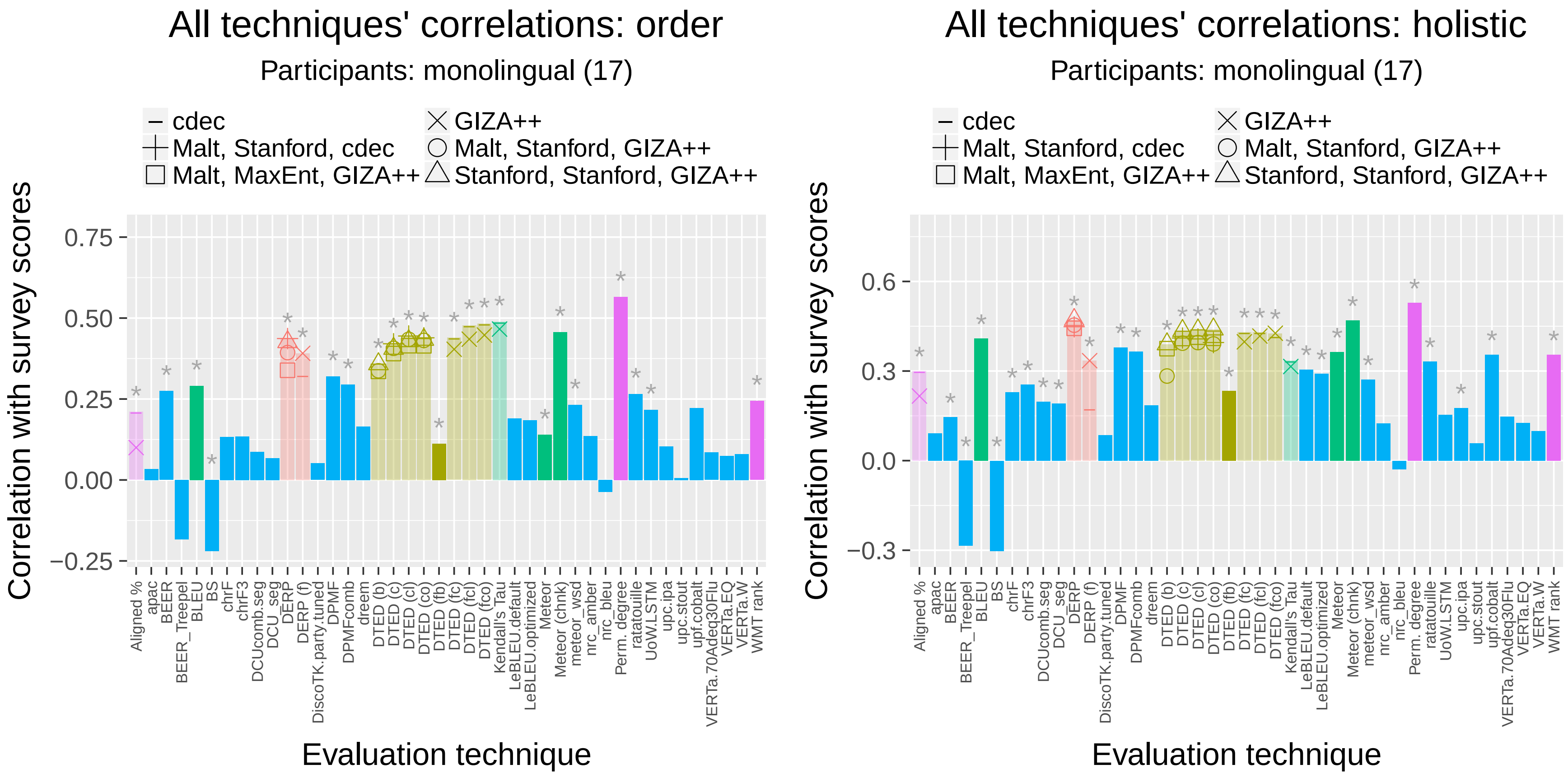


Figure B.11: Correlations between human judgments from WOJ-DB and other techniques for participants with fewer than 11 language points according to the system described in Section 6.5.2. Colours are summarised in Table 7.2 (page 117).

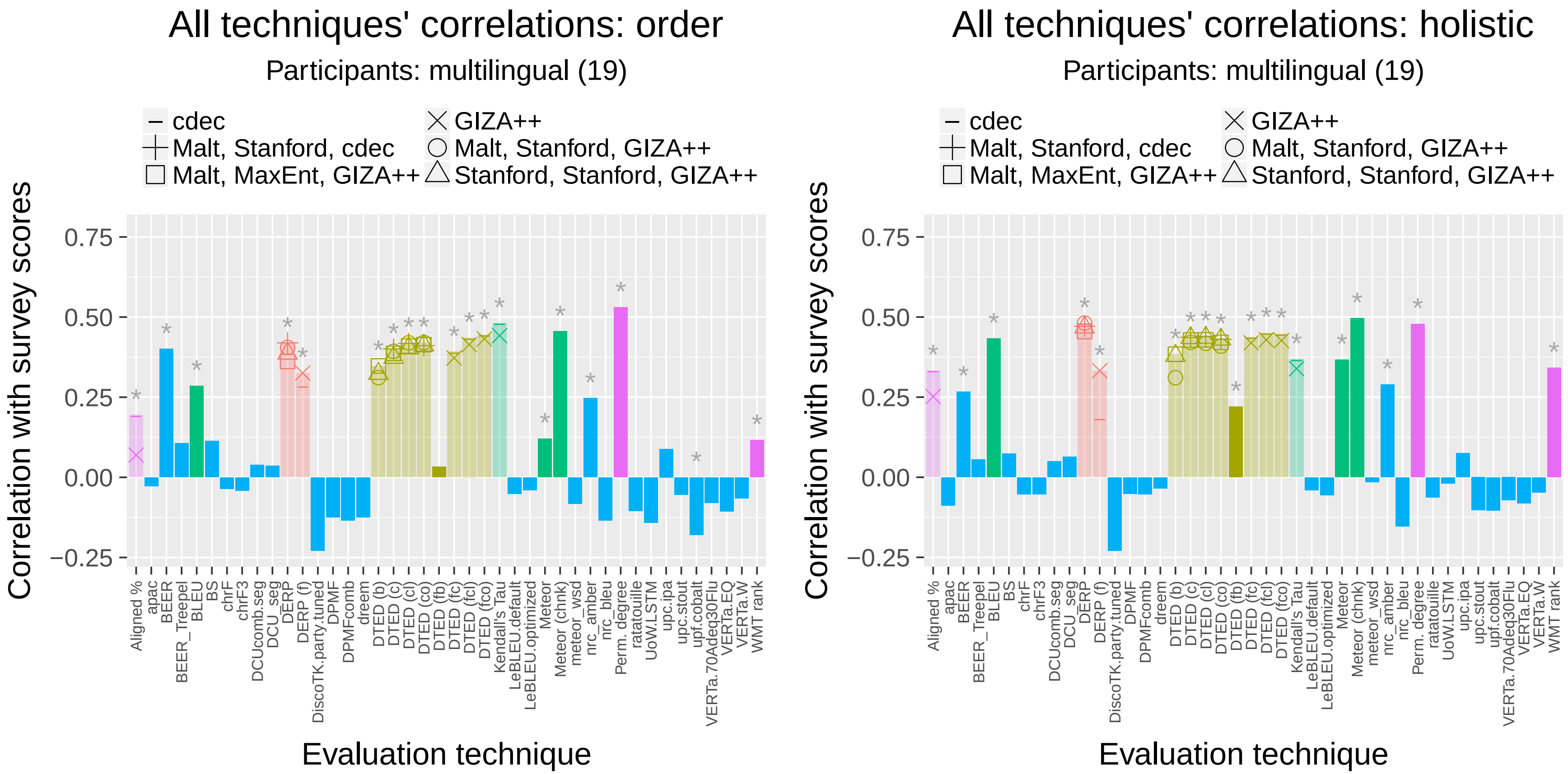


Figure B.12: Correlations between human judgments from WOJ-DB and other techniques for participants with at least 11 language points according to the system described in Section 6.5.2. Colours are summarised in Table 7.2 (page 117).

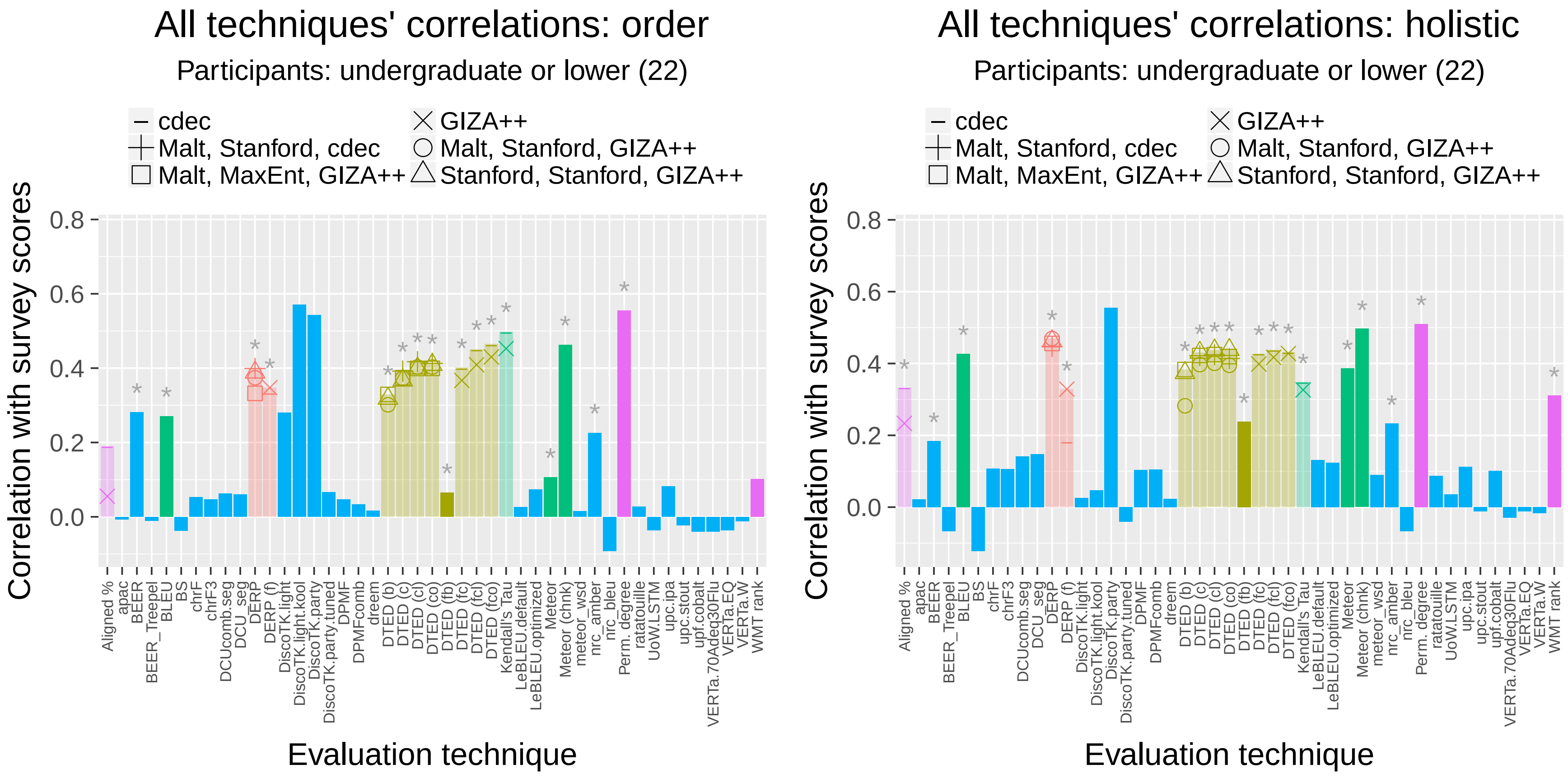


Figure B.13: Correlations between human judgments from WOJ-DB and other techniques for participants whose highest level of education was at most an undergraduate degree. Colours are summarised in Table 7.2 (page 117).

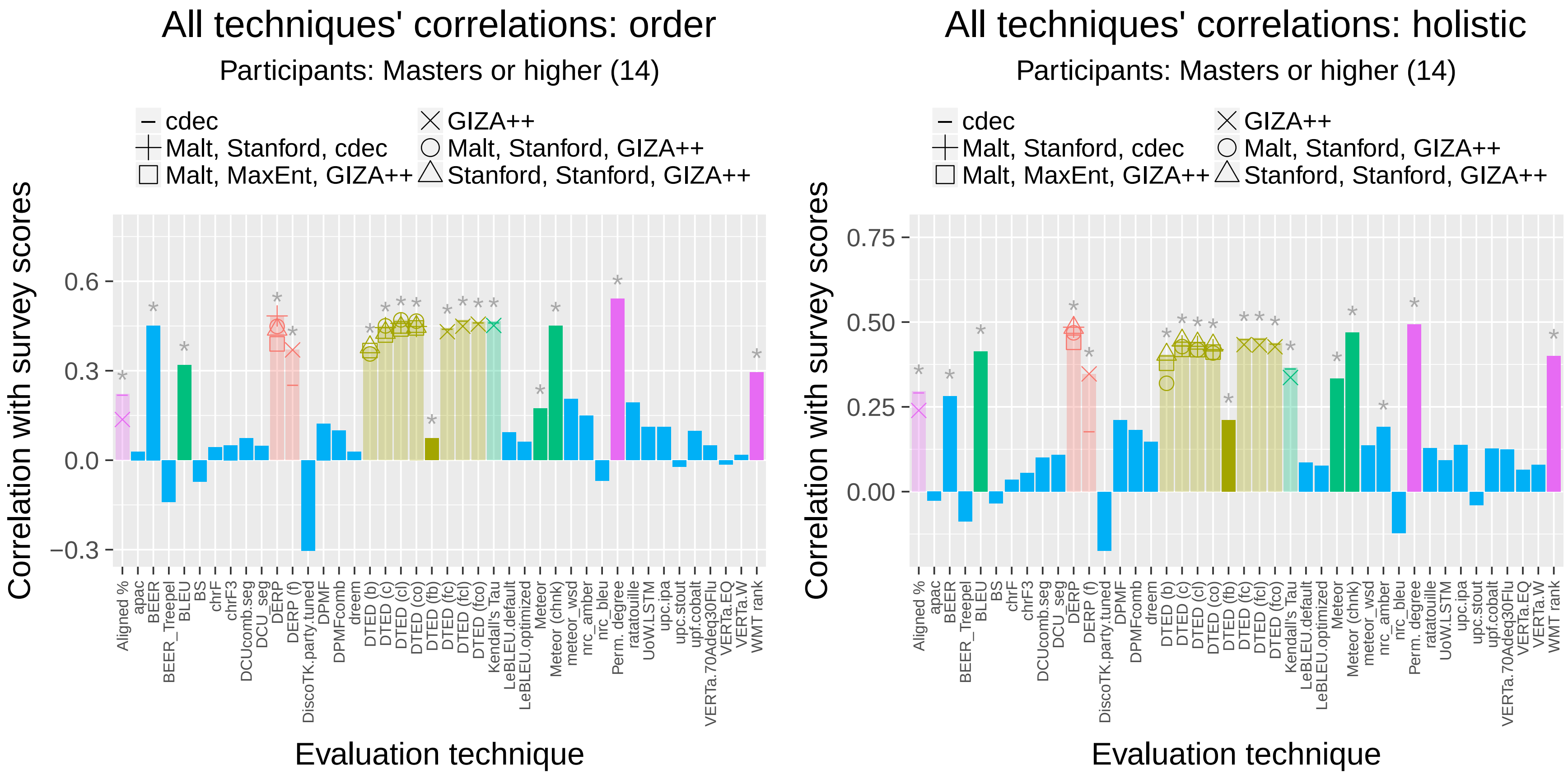


Figure B.14: Correlations between human judgments from WOJ-DB and other techniques for participants whose highest level of education was at least a Masters degree. Colours are summarised in Table 7.2 (page 117).