Multilingual Information Extraction With Machine Translation: A Pipeline Experiment Using Japanese Wikipedia

A Dissertation Presented in Partial Fulfilment of the University of Wolverhampton for the Degree of Master of Arts in Computational Linguistics

Liam O'Shaughnessy 1718055

Abstract

Information Extraction is the task of extracting machine-readable data from unstructured text. Research in this area has mostly focused on developing components for monolingual systems that extract from English text, but recently attention has turned towards multilingual systems. Machine translation has been a key technology in this area but has been criticised as impractical in its current form. This paper explores this assertion and tests how well machine-translated text is handled by information extraction components. To this end, a corpus of Japanese Wikipedia articles has been collected and translated through a state of the art cloud-based translation model. This was used as input for a proof of concept level relation extraction system utilising distant supervision techniques. Overall, the results show promise for machine translation in the information extraction pipeline, but several bottlenecks line up with the argument that it is not yet robust enough as a multilingual solution, and maybe more suited to bilingual systems.

Under the supervision of Alistair Plum
University of Wolverhampton
September 2020

Contents

List of Figures

List of Tables

Declaration

UNIVERSITY OF WOLVERHAMPTON
SCHOOL OF HUMANITIES
MA Computational Linguistics

Name: Liam O'Shaughnessy
Date: 3/10/2020
Title: Multilingual Information Extraction With Machine Translation: A Pipeline Experiment Using Japanese Wikipedia
Module Code: 7LN007
Presented in partial fulfilment of the assessment requirements for the above award
Supervisor: Alistair Plum

Declaration:
(i) This work or any part thereof has not previously been presented in any form to the University or to any other institutional body whether for assessment or for other purposes. Save for any express acknowledgements, references and/or bibliographies cited in the work, I confirm that the intellectual content of the work is the result of my own efforts and of no other person.

(ii) It is acknowledged that the author of any project work shall own the copyright. However, by submitting such copyright work for assessment, the author grants to the University a perpetual royalty-free licence to do all or any of those things referred to in section 16(i) of the Copyright Designs and Patents Act 1988 (viz: to copy work; to issue copies to the public; to perform or show or play the work in public; to broadcast the work or to make adaptation of the work).

(iii) This project did not involve direct contact with human subjects, and hence did not require approval from the LSSC Ethics Committee.

Student's signature: [illegible]
Date: 3/10/2020

1 Introduction

In 2016, a group of researchers in the journal *Trends in Ecology & Evolution* demonstrated that by 2014, the digital storage capacity of the world stood at 5 zettabytes of data, or 5×10^{12} gigabytes. (Gillings, Hilbert, and Kemp 2016). It stands to reason that as this capacity continues to grow, the capability for humans to synthesise information becomes increasingly difficult. This information is mostly made up of natural language, in the format of machine-readable text. Transforming this unstructured data into something usable for computation is a significant goal in the field of Natural Language Processing (NLP).

The task of extracting structured information from natural language data is referred to as *information extraction* (IE). More specifically this task is concerned with mapping relations between entities or events to timelines. There are exceptions, but “Generic information, conditional information, statements of knowledge, and beliefs are excluded”, the restrictions of which make output easier to interpret (Grishman 2019).

Arguably one of the key resources concerning the storage of the world’s information is the internet. The internet is essentially made up of written documents and has become a target for use in IE for extracting knowledge. However the internet is not monolingual; at the time of writing, “English is used by 60% of all the websites whose content language we know” (w3techs 2020). For this and other reasons, the development of IE systems has centered around English. However, with slower rises in performance scores, researchers are turning to the other 40% of the internet, and methods for extracting the knowledge it contains.

As content outside of English is scattered between various languages, particular attention has been paid to adapting IE systems to be multilingual where possible. One key technology that has been utilised is *Machine Translation* (MT), which has a lot of components in common with IE. The issue, however, is that MT has historically been error-prone, especially before the introduction of neural MT, and there remain arguments that until this changes, it is unsuited for use in adapting IE as a truly multilingual task, found for example in Claro et al. 2019.

This paper explores the aspects of this argument, and assesses how they hold up using state of the art methods such as neural machine translation, and increased resources for high-quality open source models. I further hypothesised that improvements in IE may now extract even from low-quality machine-translated text. To elaborate, three main research questions will be tested:

1. With state of the art methods, is MT viable in the IE pipeline?
2. With recent advancements in IE, can information be extracted from even low-quality machine-translated text?
3. How suitable is MT as a multilingual solution to IE?

To this end, an experiment was conducted using machine-translated Japanese Wikipedia articles that were used as input for an IE pipeline. The hope was that a proof of concept would allow me to explore the above research questions, with scope to expand the system to include other languages. With this, the test could highlight what works with regards to multilingual IE, and reveal areas of improvement for future research, as the literature in this area remains scarce.

As an overview, I first explore the literature surrounding IE and its main components to build a foundation, before demonstrating how systems have been adapted to a multilingual dimension. This is done throughout section 2, and after tying together the main areas of research, I describe the rationale for the experiment presented in this paper in more detail. Section 3 contains the methodology for the experiment and describes the decisions that were made as guided by the literature. Section 4 describes the results from the models trained for this experiment, and section 5 contains a description of the significance of these results, plus some further analysis of the experiment as a whole. A conclusion and further improvements for future work are then described in section 6.

2 Literature Review

2.1 Definitions and Overview

I have already given a brief definition of IE, but before investigating the literature, I will also highlight key terms and definitions of pipeline components. The first task involves finding and labelling all entities in a text, known as *named entity recognition (NER)*. The next step is to link variations of discovered entities using *coreference resolution*, which is defined as "linking two or more expressions that are used to refer to the same discourse entity" (D. Jurafsky and Martin 2019b).

Relation extraction or *event extraction* follow depending on the goal. Relation extraction is defined as the task of "finding and classifying semantic relations among text entities", and are often binary relations like 'entity 1' child-of, employed-by, or part-of 'entity 2' (D. Jurafsky and Martin 2019a). *Event extraction* is defined as finding events in which entities participate (D. Jurafsky and Martin 2019a).

It is worth noting that although IE is made up of many components, it can be a component of larger NLP systems. Its components can also be used interchangeably with other task systems. For example, IE could be confused with *information retrieval (IR)*. Whereas IE deals with extracting knowledge from text, information retrieval classifies, and retrieves entire documents that match a topic query. *Automatic Text Summarisation* may also seem similar but preserves the meaning of a document, rather than specifying information types in a structured way (Masche 2004). It also benefits from the similar classification task of finding text that contains relevant information (Masche 2004).

Another example is *Question Answering (QA)*. This focuses on systems that automatically answer factoid questions. (D. Jurafsky and Martin 2019c). This is true especially in the case of *knowledge-based QA*, which involves building a semantic representation of a query. An example given is "What states border Texas" which needs to be represented as 'entity 1', borders, 'Texas' (D. Jurafsky and Martin 2019c). This task relies heavily on relation extraction and is why building knowledge bases, also done with IE, is important.

There are numerous applications of IE outside the NLP domain. For example, ding2015deep performs event extraction on news text to predict the stock market. In Y. Wang et al. 2018, IE was performed on electronic health records to support automated medical systems. In the science domain, (Peters et al. 2014), describe their system 'PaleoDeepDive', that extracts data from publications to “describe the geological history of taxonomic diversity and genus-level rates of origination and extinction”. A salient example that incorporates IE, QA, and IR, is internet search engines. IE research, as stated, has mostly focused on English, but with 40% of web pages written in all other languages, a new dimension of IE is being considered for this and other multilingual environments.

Research regarding the extraction of information from multilingual sources is referred to as *multilingual information extraction*. As already mentioned, this research has leaned heavily on MT (Masche 2004), but the question of its viability in the IE pipeline is still being contested (Claro et al. 2019).

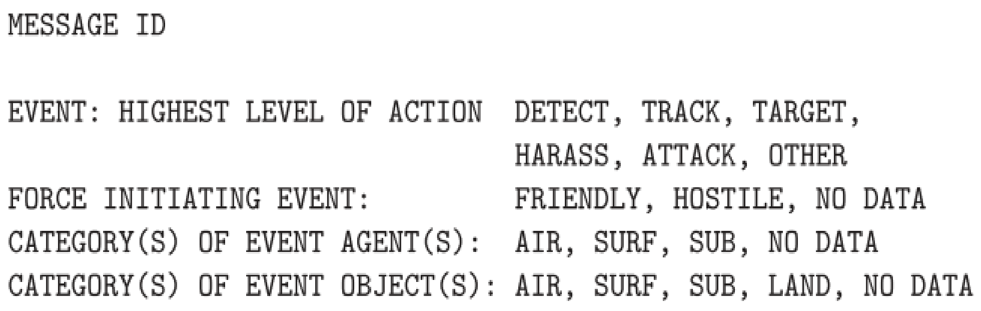
To explore the literature surrounding these concerns, two key sources already referred to will be consulted; Grishman 2019, and D. Jurafsky and Martin 2019a. These sources are recent and provide an in-depth overview of IE. As such, critical exploration and comparison can arguably shape a basal understanding of each topic. From here, relevant sources can be interwoven to fill in any gaps, further understanding, and provide a rationale for the practical elements presented after chapter 2.

2.2 Rule Based Systems and Evaluation

To begin, it is worth noting the differences between Grishman 2019 and D. Jurafsky and Martin 2019a. Grishman’s article focuses on how IE has changed historically and is therefore more succinct. D. Jurafsky and Martin 2019a is found in *Speech and Language Processing*. This is more empirical and is therefore more descriptive. Both sources highlight similar topics, and both discuss slightly varying points. Further, both sources miss out on discussions surrounding multilingual methods which will be discussed.

Jurafsky and Martin begin by highlighting that IE developed from research into human knowledge structures (Schank and Abelson 1977). This describes that events found in written text can be mapped to prototypical sequences of sub-events enacted by expected participants. This is referred to as *scripts* (D. Jurafsky and Martin 2019a). These scripts can “facilitate the proper classification of entities” and assign entities into “roles and relations”. They can also be represented as *templates* with fixed slots. Research in this area eventually lead to a series of conferences that began in 1988 concerned with automatically extracting facts from documents to fill templates, and were known as the *message understanding conferences (MUCs)* (Grishman and Sundheim 1996). Grishman’s survey begins here with MUCs 1-3 which began to define the task of filling template slots, as well as how this would be evaluated (Grishman 2019).

MUCs were funded by the US government and initially involved Navy reports. The first template contained 10 slots to be filled from this input. The following figure taken from (Grishman 2019) provides an example of an unfilled template.



```
MESSAGE ID

EVENT: HIGHEST LEVEL OF ACTION  DETECT, TRACK, TARGET,
                                HARASS, ATTACK, OTHER
FORCE INITIATING EVENT:         FRIENDLY, HOSTILE, NO DATA
CATEGORY(S) OF EVENT AGENT(S):  AIR, SURF, SUB, NO DATA
CATEGORY(S) OF EVENT OBJECT(S): AIR, SURF, SUB, LAND, NO DATA
```

Figure 1: Example from MUC-2 Template.

By MUC-3 the methods used to evaluate template filling systems were decided to be precision, recall, and a harmonic mean of the 2, known as F_1 score. These methods were not initially intuitive, but interestingly "have served as a model for evaluations in many other areas of NLP" (Grishman 2019). The general way to calculate precision is:

$$precision = \frac{TruePositives}{TruePositives + FalsePositives}$$

In the context of MUC, this was interpreted as: out of all slots filled by a system, how many were correct?

$$precision = \frac{CorrectlyFilled}{CorrectlyFilled + IncorrectlyFilled}$$

This was possible as comparison keys that were manually filled were provided. Modern IE tasks where annotation is not available often need to be calculated by checking a sample of the output.

Recall gives a measure of how many true positives out of all true positives were found by a system.

$$recall = \frac{TruePositives}{TruePositives + FalseNegatives}$$

In the case of template filling, false negatives do not exist, however, so recall needed to be calculated by taking the ratio of correctly filled slots and compared to the answer key. For example, if an answer key has $Nkey$ slots and a system fills $Ncorrect$ slots the following calculation is applied (Grishman and Sundheim 1996):

$$recall = \frac{Ncorrect}{Nkey}$$

Its worth noting that these metrics were not initially homogenised across the field of IE, as one might interpret from Grishman 2019. For context, Grishman demonstrates that by MUC-5, the conference had attracted international participation. Japanese became a choice alongside English extraction due to US concerns of commercial competition from Japan (Grishman 2019). Grishman does not mention any examples of other conferences participated in by these international participants, but one example is IREX *(Information Retrieval and Extraction Exercise)*. This project concluded in 1999 and took place to give a "common platform to evaluate systems with the same standard".

"We have had some difficulties comparing systems based on the same platform since our research is conducted at many different universities, companies, and laboratories using different data and evaluation measure" (Sekine and Isahara 2000).

This suggests that despite MUCs exemplifying best practices, they were not conventional across the field. Despite this, MUCs still provide good insight into IE evaluation, and how architectures and pipelines developed. Key research at this time focused on rule-based systems, with automatic systems emerging. Rule-based systems are useful to understand as components in their pipelines are still used commercially, despite being regarded as "dead-end technology in academia" (Chiticariu, Y. Li, and Reiss 2013).

A notable system referred to by the comparison sources is FASTUS. This is a template filling system that extracts information from free text using a "set of cascaded, non-deterministic finite-state transducers" (Hobbs et al. 1997). At each stage of the system, text input is processed using handwritten regular expression and grammar rules to create and apply structures that can be applied and used as input for the next level. This concept provides the basis for the IE pipeline, which in short, was that the first 4 layers "do basic tokenization, chunking, and parsing" and the final layer "extracted entities and events with an FST-based recognizer that inserts the recognized objects into the appropriate slots in templates" (D. Jurafsky and Martin 2019a). These stages were implemented for English and Japanese, but it's worth emphasizing that the system rules could not generalise, and that separate rules had to be written for each language.

The first layer of the system, after tokenization, tagged multi-words such as "joint venture", and basic entities such as names, locations, dates, and times. These were recognised by "employing specialised microgrammars" (Hobbs et al. 1997). Grishman indicates that there was a consensus that "a preprocessor to identify names was essential" (Grishman 2019). This is an example of what led to the intuition that named entity recognition should be done first in IE tasks.

An example rule-based strategy in English for this relies on using triples of a predicate and some of its arguments, such as 'subject', 'verb', 'object', but this is reported to be a problem for other languages. A language with a flexible case-marking system, like Japanese for example has a topic marker 'は' ('wa') that can mark almost any entity and can not be used to indicate a grammatical category reliably like markers in English can. There is also the issue that unlike English where a subject in a sentence is essential, in Japanese it is grammatical and common to omit the subject of a sentence as long as it can be understood from context (Sudo 2001). This zero-pronoun issue was hypothesised to also present an issue in regards to usable multilingual IE and was examined (see section 5.2). Sudo 2001 resolved these problems with structural pattern matching by parsing every sentence as a dependency tree.

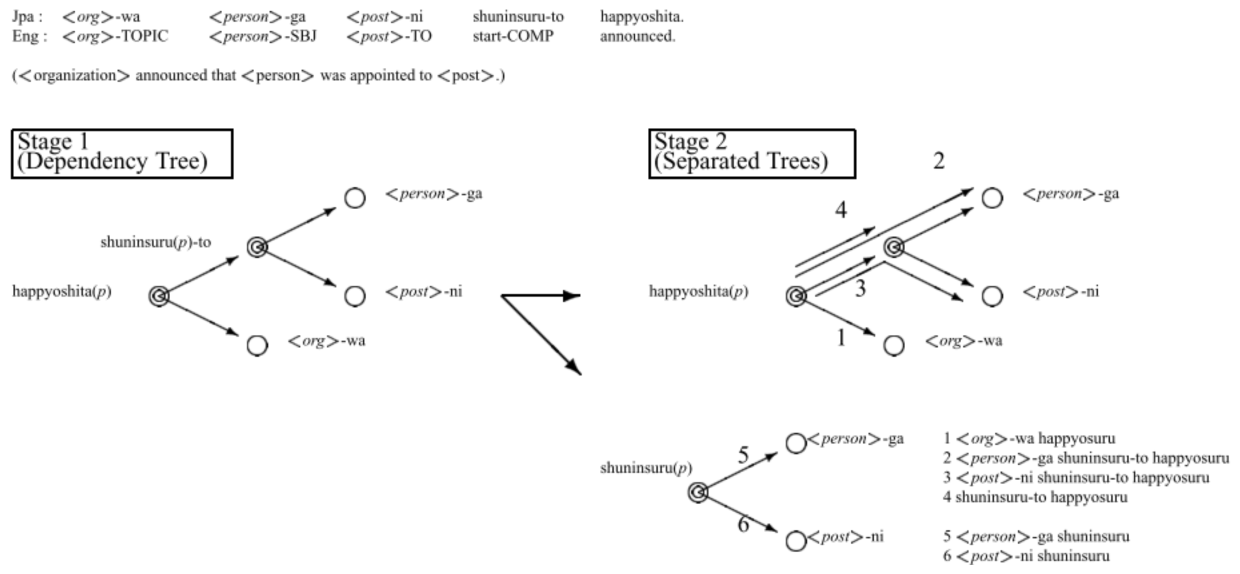

Jpa : <org>-wa <person>-ga <post>-ni shuninsuru-to happyoshita.
Eng : <org>-TOPIC <person>-SBJ <post>-TO start-COMP announced.
(<organization> announced that <person> was appointed to <post>.)
Stage 1 (Dependency Tree)
Stage 2 (Separated Trees)
shuninsuru(p)-to
happyoshita(p)
<person>-ga
<post>-ni
<org>-wa
shuninsuru(p)
1 <org>-wa happyosuru
2 <person>-ga shuninsuru-to happyosuru
3 <post>-ni shuninsuru-to happyosuru
4 shuninsuru-to happyosuru
5 <person>-ga shuninsuru
6 <post>-ni shuninsuru

Figure 2: Japanese pattern construction using dependency trees (Sudo 2001).

The FASTUS system differentiates from the methods utilised in Sudo 2001. Grishman notes that full sentence parsing was a topic of disagreement amongst MUC participants, as full parse trees defining every word in a sentence were difficult to do and computationally expensive at the time (Grishman 2019). Hobbs et al. 1997 were able to use a method called *chunking* as an alternative, which was faster but only provided partial structure. Specifically, Stage 2 identified noun groups, verb groups, and "critical word classes including prepositions, conjunctions, relative pronouns, and the words 'ago' and 'that'", and discarded phrases subsumed by larger phrases (Hobbs et al. 1997). Stage 3 then used that input to parse more complex noun and verb groups. This allowed for stage 5 to apply non-deterministic rules that could skip groups in conjunction with trigger words and solved the problem of extracting information from a sentence expressing two events. Grishman gives the example:

person relativePronoun (nounGroup — other)* verbGroup (nounGroup — other)* resigned

which matches a sentence like "Fred, who was named president of IBM last year, suddenly resigned yesterday" (Grishman 2019).

Despite the benefits offered by systems like FASTUS, the literature does point towards full parsing being the preferred method. Accurate parsers trained using the Penn Treebank[1] made full-sentence parsing more viable, and let researchers introduce automatic pattern extracting methods that relied on their accuracy (Grishman 2019). These methods were

[1]The Penn Treebank is a treebank that parses text for syntactic and semantic information

much less labour intensive than handwritten rules, and by the end of MUC, gave comparable results. For example, as shown in Sudo 2001, the authors focused on automatic pattern extraction for Japanese, using MUC style template filling, and report the scores: recall (42%), and precision (65%). Compare this to the results achieved by FASTUS in the Japanese MUC-5 task: recall (34%), and precision (62%) (Hobbs et al. 1997). These results are comparable to systems found in the previously mentioned IREX project, which is also useful to compare as this was performed on Japanese text. Although the focus was on named entity extraction from news corpora, rather than template filling. The median F-score presented at IREX was 58% and the paper describes that the scores were from a mix of handwritten pattern systems, automatic pattern extraction systems, and fully automatic systems. It is further reported that "the top three systems came from these three categories", and that they could not "simply conclude which type of systems outperformed others" (Sekine and Isahara 2000).

IREX demonstrates several things that are also highlighted in Grishman's historical accounting. First, that machine learning methods were starting to be implemented by NLP researchers at around this time, and that rather than full systems, focus on specialising in subsystems began taking precedence, such as NE extraction in IREX. This can be seen in MUC-7 which offered *NE extraction*, *coreference resolution*, *template element*, and *template relation* tasks (Grishman 2019). The scores presented in the given example papers were also seen as "topped out" by this point. As well as this, researchers began to assert that templates "led to a lot of work not directly relevant to IE" (Grishman 2019). This, in conjunction with the rise of the internet and subsequent increase in electronic information, led to a desire to move away from just taking information "off the page" and moving towards "developing the capability to extract meaning from sources" (Doddington et al. 2004). Thus the decision was made to end MUC and succeed it with what became the *Automatic Content Extraction (ACE)* program.

2.3 Named Entity Recognition

The task "extracting meaning" was "identified in general as the extraction of the entities, relations, and events being discussed in the language" (Doddington et al. 2004). These core extraction tasks enable research to focus on developing "core enabling technologies", and finding objects in text, rather than being concerned with the words in text.

"For example, the so-called 'named entity' task, as defined in MUC, is to identify those words (on the page) that are names of entities. In ACE, on the other hand, the corresponding task is to identify the entity so named. This is a different task, one that is more abstract and that involves inference more explicitly in producing an answer" (Doddington et al. 2004).

Doddington et al. 2004 defines the 3 ACE tasks in detail, and its clear from the literature that while there has been some branching out, these tasks are still considered core to IE. The next step in the review then is to compare how these tasks are assessed by both Grishman 2019 and D. Jurafsky and Martin 2019a, and present the relevant literature connected to them.

Its from this point that more attention can be turned to D. Jurafsky and Martin 2019a. As described, both texts outline the consensus in the literature that NER is done first, or performed as a component of other applications. This is also the first task identified in ACE, and is described in the program as "Recognition of entities, not just names", and details that an entity, whether a name, a description, or a pronoun, are to be found and collected into equivalence classes based on reference to the same entity (Doddington et al. 2004). ACE defined these classes, used to label types of entities when recognised as such:

Type	Examples
person	Fred Smith: the undertaker
organization	Ford; San Francisco 49ers; a car manufacturer
GPE	France; Los Angeles
location	Nile; Mt. Everest; southern Africa
facility	Disneyland; the Berlin Wall; Aden's streets
vehicle	the U.S.S. Cole; the train; the helicopter
weapon	Anthrax; bullets; tear gas

Table 1: ACE entities (Grishman 2019).

Its clear, as noted by Jurafsky and Martin, that these entities will not cover every task and many applications need to predefine specific entity types such proteins, genes, commercial products, or works of art (D. Jurafsky and Martin 2019a). However, it is clear from the literature that the ACE classes were refined and became generalised labels used by applications. The least fine-grained labels, for example, currently used by *spaCy*[2] are shown in table 2.

[2]spaCy is an open-source library of software used for various natural language processing tasks.

Type	Description
PER	Named person or family.
LOC	Name of politically or geographically defined location.
ORG	Named corporate, governmental, or other organizational entity.
MISC	Miscellaneous entities, e.g. events, nationalities, products or works of art.

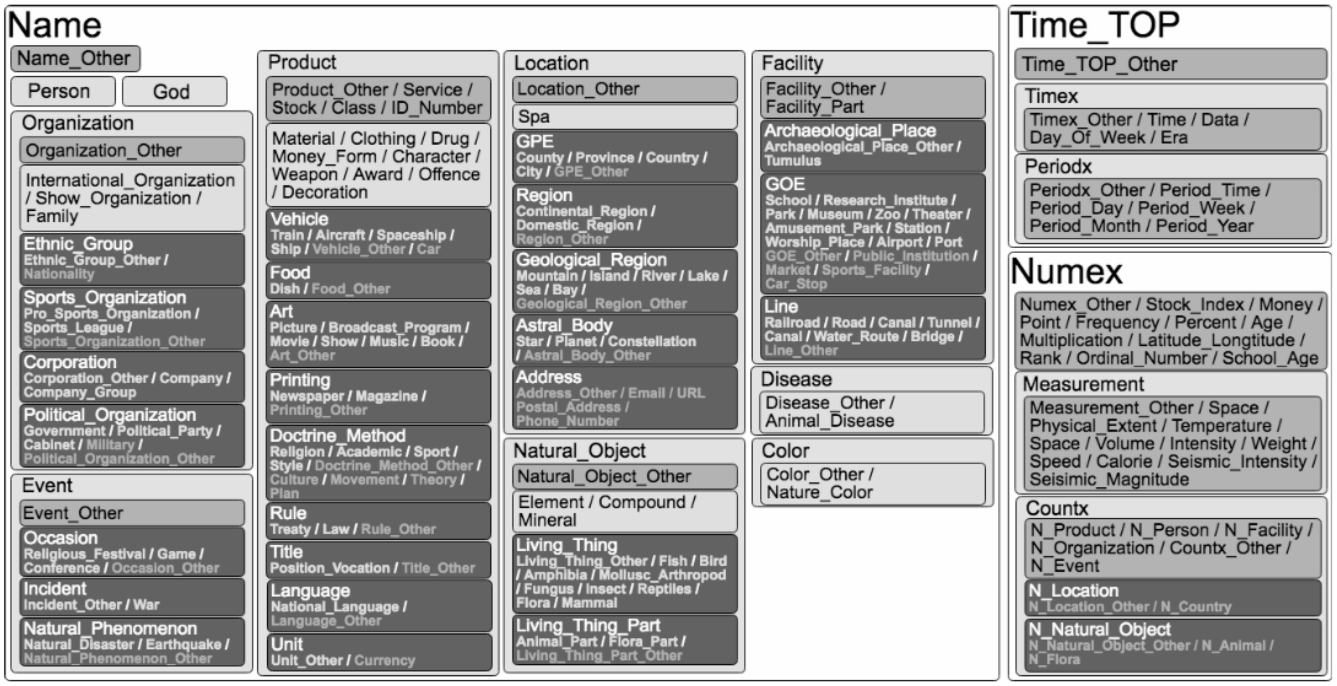
Table 2: Example of spaCy NER annotations (ExplosionAI 2020).

One might note some difficulties that arise from this, the first being type ambiguity. In the ACE classes (table 1), GPE (geopolitical entity) is defined as a territory that contains a government structure, but it is also possible to note these as locations. This ambiguity taken to an extreme is demonstrated in table 3. This is useful to be aware of when defining an IE task and was a consideration in deciding on the information to be extracted to keep the experiment presented in the paper as simple as possible.

[PER Washington] was born into slavery on the farm of James Burroughs.
[ORG Washington] went up 2 games to 1 in the four-game series.
Blair arrived in [LOC Washington] for what may well be his last state visit.
In June, [GPE Washington] passed a primary seatbelt law.
The [VEH Washington] had proved to be a leaky ship, every passage I made...

Table 3: Examples of NER type ambiguities taken from (D. Jurafsky and Martin 2019a).

A recent ongoing project known as SHINRA explores the problems with ambiguity and label definitions, but this was secondary to its main purpose (Sekine, Kobayashi, and Nakayama 2018). Here it is asserted that knowledge base construction, and by extension NER, has suffered from the incorporation of too much noise and inconsistency by databases being constructed with a "bottom-up approach". Wikidata for example, used in this paper (section 3.3), categorises entities inconsistently such as "like a city", "city/town" etc. The purported solution is to use a "top-down" approach and design an ontology for named entities. The SHINRA team uses Sekine's Extended Named Entity (ENE) classification hierarchy to do this, which includes 200 categories under 4 hierarchical layers (Sekine, Kobayashi, and Nakayama 2018). It will be interesting to observe what impact the project has on the field, but currently, it demonstrates a problem when comparing metrics for NER systems, which may differ depending on how fine-grained labels are.



Name
Name_Other
Person
God
Organization
Organization_Other
International_Organization / Show_Organization / Family
Ethnic_Group
Ethnic_Group_Other / Nationality
Sports_Organization
Pro_Sports_Organization / Sports_League / Sports_Organization_Other
Corporation
Corporation_Other / Company / Company_Group
Political_Organization
Government / Political_Party / Cabinet / Military / Political_Organization_Other
Event
Event_Other
Occasion
Religious_Festival / Game / Conference / Occasion_Other
Incident
Incident_Other / War
Natural_Phenomenon
Natural_Disaster / Earthquake / Natural_Phenomenon_Other
Product
Product_Other / Service / Stock / Class / ID_Number
Material / Clothing / Drug / Money_Form / Character / Weapon / Award / Offence / Decoration
Vehicle
Train / Aircraft / Spaceship / Ship / Vehicle_Other / Car
Food
Dish / Food_Other
Art
Picture / Broadcast_Program / Movie / Show / Music / Book / Art_Other
Printing
Newspaper / Magazine / Printing_Other
Doctrine_Method
Religion / Academic / Sport / Style / Doctrine_Method_Other / Culture / Movement / Theory / Plan
Rule
Treaty / Law / Rule_Other
Title
Position_Vocation / Title_Other
Language
National_Language / Language_Other
Unit
Unit_Other / Currency
Location
Location_Other
Spa
GPE
County / Province / Country / City / GPE_Other
Region
Continental_Region / Domestic_Region / Region_Other
Geological_Region
Mountain / Island / River / Lake / Sea / Bay / Geological_Region_Other
Astral_Body
Star / Planet / Constellation / Astral_Body_Other
Address
Address_Other / Email / URL Postal_Address / Phone_Number
Natural_Object
Natural_Object_Other
Element / Compound / Mineral
Living_Thing
Living_Thing_Other / Fish / Bird / Amphibia / Mollusc_Arthropod / Fungus / Insect / Reptiles / Flora / Mammal
Living_Thing_Part
Animal_Part / Flora_Part / Living_Thing_Part_Other
Facility
Facility_Other / Facility_Part
Archaeological_Place
Archaeological_Place_Other / Tumulus
GOE
School / Research_Institute / Park / Museum / Zoo / Theater / Amusement_Park / Station / Worship_Place / Airport / Port GOE_Other / Public_Institution / Market / Sports_Facility / Car_Stop
Line
Railroad / Road / Canal / Tunnel / Canal / Water_Route / Bridge / Line_Other
Disease
Disease_Other / Animal_Disease
Color
Color_Other / Nature_Color
Time_TOP
Time_TOP_Other
Timex
Timex_Other / Time / Data / Day_Of_Week / Era
Periodx
Periodx_Other / Period_Time / Period_Day / Period_Week / Period_Month / Period_Year
Numex
Numex_Other / Stock_Index / Money / Point / Frequency / Percent / Age / Multiplication / Latitude_Longtitude / Rank / Ordinal_Number / School_Age
Measurement
Measurement_Other / Space / Physical_Extent / Temperature / Space / Volume / Intensity / Weight / Speed / Calorie / Seismic_Intensity / Seisimic_Magnitude
Countx
N_Product / N_Person / N_Facility / N_Organization / Countx_Other / N_Event
N_Location
N_Location_Other / N_Country
N_Natural_Object
N_Natural_Object_Other / N_Animal / N_Flora

Figure 3: Sekine's Extended Named Entity Hierarchy (Sekine, Kobayashi, and Nakayama 2018).

The task itself as laid out in ACE was recognised as a token sequence labelling problem that could be solved most effectively with *Maximum Entropy Markov Models* (MEMM)[3] or *Conditional Random Field* (CRF)[4] models (Grishman 2019) (D. Jurafsky and Martin 2019a). Other supervised learning methods include *Support Vector Machines (SVM)* or Decision Trees, such as used in Sudo 2001.

These methods all utilise annotated corpora, lists of entities such as gazetteers[5] and extracting discriminative features to create disambiguation rules. Common features used with these classifiers are related to word shape, such as representing the pattern of a word by mapping word case and length. (Nadeau and Sekine 2007). Morphology is also frequently highlighted as important, including prefix, suffix singular version, and stem information (Nadeau and Sekine 2007) (D. Jurafsky and Martin 2019a) (Grishman 2019).

The issue, as shown in other parts of the pipeline that utilise supervised methods is

[3]A Maximum Entropy Markov Model is a sequence labelling model that combines hidden Markov model features with multinomial logistic regression

[4]A Conditional Random Field is a classification model where predictions are made based on contextual information

[5]A gazetteer is a geographical dictionary that provides a list of place names and associated information

that annotating a large corpus, and even feature extraction is laborious. Therefore neural networks that can capture arbitrary functions from data have succeeded these methods. The trade-off being that the data requirements and training times are larger. The method used for NER is a *Bi-Directional Long Short Term Memory (Bi-LSTM)*[6] model in combination with a CRF layer (D. Jurafsky and Martin 2019a)(Grishman 2019). Grishman notes the best performance "is currently obtained by combining a dual token/character model with contextualized word embeddings" (Grishman 2019). This method addresses the weakness of character-level contextualised models when encountering rare words with no context, by intuiting that entities given no context are expected to be known to the reader. This allows the authors to aggregate the contextualised word embeddings of each rare occurrence string, and then use a global word representation to give each occurrence of a rare word in the dataset a new embedding (Akbik, Bergmann, and Vollgraf 2019). F-scores of up to 93.18% are recorded for these methods.

Coreference resolution is also considered by the comparison sources as useful before NER. This is a dense topic that I have opted to exclude from the scope of this paper, but a summary explains that the main models used include a *mention-mention model*, that classifies pairs of entities, provides a probability of coreference and then resolves conflicts. Also, *mention-entity models* pass over a document and assign a mention of an entity to either a previously mentioned entity or create a new one (Grishman 2019).

2.4 Relation Extraction

The next step in "extracting meaning" as defined by ACE involves the "detection and categorisation of relations between pairs of entities". These were represented "in terms of their attributes and their (two) arguments". The arguments are the ACE entities that are related, and the attributes are the relation type (Doddington et al. 2004). There were five general types of relations decided, some of which were further divided for a total of 24 types. The broad categories of these types include:

- Role - The role a person plays in an organisation
- Part - part-of relationships

[6] A Bi-Directional long short term memory model generates word and character embeddings by combining the predictions of a recurrent neural network taking the input as a sequence from left to right and another taking it as right to left which allows a model to utilise future information

- At - location relationships
- Near - relative location relationships
- Social - Parent, sibling, spouse, etc

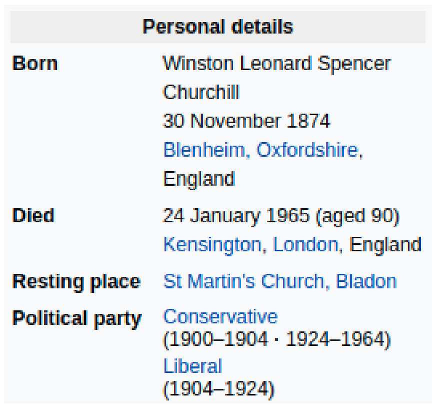
(Doddington et al. 2004)

Examples are presented in table 4. In its simplest form, this is a classification problem that predicts a type of relation (or no relation) between each entity pair in a corpus.

Relations	Types	Examples
Physical-Located	PER-GPE	[He] was in [Tennessee]
Part-Whole-Subsidiary	ORG-ORG	[XYZ], the parent company of [ABC]
Person-Social-Family	PER-PER	[Yoko]'s husband [John]
Org-AFF-Founder	PER-ORG	[Steve Jobs], co-founder of [Apple]...

Table 4: Examples of relations between named entities (D. Jurafsky and Martin 2019a).

Jurafsky and Martin allude to the issue of domain defined relation sets, similar to the issue presented in defining NER types. This is not mentioned in Grishman 2019. ACE relations can only be used in specific contexts and different sets must be defined depending on the task. For example as discussed in Sekine, Kobayashi, and Nakayama 2018, knowledge bases such as Wikidata supply numerous amounts of attributes of entities that are essentially built from defining relation types from Wikipedia infoboxes. A sample is provided below:



Personal details	
Born	Winston Leonard Spencer Churchill 30 November 1874 Blenheim, Oxfordshire, England
Died	24 January 1965 (aged 90) Kensington, London, England
Resting place	St Martin's Church, Bladon
Political party	Conservative (1900–1904 · 1924–1964) Liberal (1904–1924)

Figure 4: Sample of infobox from Winston Churchill Wikipedia page (Wikipedia 2020).

Such infoboxes could be constructed from relation tuples extracted from text such as ('Winston Churchill', date-of-death, '24-01-1965'). The problem again is one of ambiguous definitions when constructing machine-readable databases.

Both comparison sources indicate supervised learning methods were a common approach, and this remains a predominant method in the literature. It is usual to work at the sentence level, and after finding entities, a filtering binary classifier can be utilised to determine first if there is a relation or no relation. After extracting positive and negative examples, another classifier can assign labels to the positive sentences. (D. Jurafsky and Martin 2019a)(Grishman 2019). As usual, this method used annotated corpora and feature extraction.

Several common features defined for this task include the following: entity types of both entities given as arguments, bag-of-words, and bigrams where arguments are distinguished from non-argument tokens, grammar productions, and dependency relations represented as a dependency path between two entity arguments (J. Jiang and Zhai 2007, pp. 116–117).

The same supervised classification task can also utilise neural models without the need for feature extraction, such as a bi-LSTM with word embeddings as inputs, such as used for NER. However, it is noted that due to relations holding between entities even across sentences, it is possible to use other algorithms such as CNNs or tree LSTMs (D. Jurafsky and Martin 2019a). Using neural models, however, still does not offset the arduous annotation phase that remains an issue for supervised learning. One may also notice that, as mentioned, using predefined labels leads to classifiers being biased towards specific domains, which limits extraction potential. For this reason, attention was turned instead to semi-supervised and unsupervised methods.

The first semi-supervised method is called *bootstrapping* and works by taking a small amount of 'seed tuples' (2 entities as arguments labelled with a positive relation) and searching a corpus for sentences containing both arguments in the seed. It is then possible to generate patterns based on these sentences to extract more sentences with different entity relations, or new sentences that can be used to find more patterns to search with. This iterative system has been used effectively but has been reported to result in low precision and semantic drift, which is when erroneous patterns introduce erroneous tuples. (Mintz et al. 2009).

This concept was ameliorated in Snow, Jurafsky, and Ng 2005, where WordNet[7] is used to

[7]WordNet is a lexical database of semantic relations between words (Fellbaum 2012)

create seed tuples of entities that have hypernym/hyponym (is-a) relations, such as ('Shakespeare', is-an, 'author'). After generating patterns from sentences found using the seed tuples, like with bootstrapping, the sentences were parsed for pattern features and used to train a hypernym classifier that could be used to replace the handwritten database that existed. This method was reported to improve on WordNet-based classifiers by a 40% relative F-score (Snow, Jurafsky, and Ng 2005). More significantly, however, was the implication that supervised learning could be performed without the need for hand annotation.

The technique of *distant supervision* was developed based on this paradigm (Mintz et al. 2009). Instead of using a small number of seeds, here Freebase[8] was used to extract large numbers of sentences from 1.2 million Wikipedia articles with the intuition that "any sentence that contains a pair of entities that participate in a known Freebase relation is likely to express that relation in some way" (Mintz et al. 2009). This allows a database of facts to be converted to an annotated corpus to use for supervised classifiers. "Suppose we have a database with a relation R consisting of pairs $< x1, y1 >, < x2, y2 >, ...$, and that some of these pairs appear in the corpus separated by word sequences Wi. We will annotate every sequence Wi as expressing relation R" (Grishman 2019).

After aggregating syntactic and lexical features, possible due to a large amount of data, a 67.6% precision score was achieved along with 10,000 instances of 102 relations being extracted, including relations not found in Freebase. It was also notable that Wikipedia was used "because its sentences tend to make explicit many facts that might be omitted in newswire" (Mintz et al. 2009). When considering multilingual methods it was worth considering this paradigm because it avoids domain dependence, and the short structured sentences of Wikipedia articles were hypothesised to suit MT.

The final task of unsupervised relation extraction, used when there is no labelled data or external lists of relations to rely on, is known as *Open IE*. This task is distinguished from general IE given its goal of reducing written text to simple sentences in the form of subject-verb-object tuples (Grishman 2019). Due to this task differentiation, literature regarding Open IE will be omitted. However, for the sake of analysis D. Jurafsky and Martin 2019a explore this topic in more detail whereas Grishman 2019 includes this same cursory information, perhaps to present a more focused exploration.

[8]Freebase was a large knowledge base composed of data from many sources, described as "an open shared database of the world's knowledge" (Bollacker et al. 2008). It was succeeded by Wikidata

2.5 Event Extraction

Event extraction came later in the ACE evaluations, and task definition and annotation guidelines were not initially included (Doddington et al. 2004). Eventually, it was decided that ACE events, like entities and relations, would fall under a range of types. 8 main types divided into sub-types for a total of 34 categories were developed. In addition, events had 4 other attributes:

1. Modality (Asserted, Other)
2. Polarity (Positive, Negative)
3. Genericity (Specific, Generic)
4. Tense (Past, Present, Future, Unspecified)

(Ahn 2006).

Type	Examples
life	is born; marries; dies
movement	transport; travel
transaction	sell; purchase; acquire
business	found; merge
conflict	attack; demonstrate
contact	meet; phone; write
personell	hired; fired; elected
justice	arrest; trial; convict

Table 5: 8 Main ACE event types taken from (Grishman 2019).

It is noteworthy that the ACE model of event extraction was not the only approach to the task. The ACE model treats an event as a complex structure with relating arguments that are also complex structures. Also, only events that fall under the predefined categories are annotated. The other approach is known as the TimeML model, in which all events in a text are annotated and represented by "a word that points to a node in a network of temporal relations" (Ahn 2006). Example annotation is shown below for both models (D. Jurafsky and Martin 2019a).

```
[EVENT Citing] high fuel prices, United Airlines [EVENT said] Friday
it has [EVENT increased] fares by $6 per round trip on flights to some
cities also served by lower-cost carriers.

A fare increase initiated <TIMEX3> last week</TIMEX3> by UAL Corp's
United Airlines was matched by competitors over
<TIMEX3>the weekend</TIMEX3>, marking the second successful fare
increase in <TIMEX3>two weeks</TIMEX3>.
```

TimeML annotation is useful to consider, as temporal expressions are important across all tasks in IE. Out of the comparison sources, only D. Jurafsky and Martin 2019a addresses this. These expressions can be categorised 3 ways: as *absolute expressions* that can be mapped onto a calendar ('4th October 2020'), as *relative expressions* that map to times through a reference point (such as 'a week from tomorrow'), and *durations*, which represent a time span ('for 2 days') (D. Jurafsky and Martin 2019a). The lexical triggers that denote temporal expressions are useful, as they overlap with the first step of event extraction, which is anchor identification (Ahn 2006). Temporal normalisation for expressions may also be useful when considering multilingual IE. For example, the way dates are written vary by language, so mapping dates to the ISO 8601 standard (ISO8601 2004) ('2020-10-04T18:16:28+00:00') during preprocessing may be beneficial. Temporal expressions are entities and can be extracted using sequence labelling methods already discussed in section 2.3.

Event extraction was initially solved with a modular approach. Ahn 2006 divided the task into stages that utilised different machine learning classifiers. The first stage, as mentioned was finding anchors and marking them as events. Next, event arguments are identified (timeexes, entity mentions, and values). The third stage handled determining modality, polarity, genericity, and tense attributes. Finally, coreference resolution determined which events were the same (Ahn 2006). Grishman helpfully notes that the problem with the modular approach is accuracy loss, as anchors and arguments are predicted in isolation, and anchors such as verbs are ambiguous without their arguments. "Firing a person is a different type of event than firing a rocket" (Grishman 2019). He also notes that a better solution is joint inference. This was proposed by Q. Li, Ji, and Huang 2013 by presenting an algorithm to predict triggers and arguments simultaneously, the intuition being that adjacent events often co-occur and share the same arguments. Example sentence provided (Q. Li, Ji, and Huang 2013, pp. 73–74):

```
'In Baghdad, a cameraman [EVENT died] when an American tank
```

`[EVENT fired] on the Palestine Hotel'.`

This paradigm initially used a combination of local features (trigger features and argument features) and global features from text spans denoting entire events, with a structured perceptron[9] to train the model. F_1-scores of 70.4% (trigger identification), and 56.8% (argument identification) were reported, which surpassed the previous gold standard.

Concerning deep learning methods, joint inference has been used with bidirectional recurrent neural networks. Nguyen, Cho, and Grishman 2016 for example, raise the F-score to 73.3% (trigger identification) and 62.8% (argument identification) by employing this method, which essentially has a richer representation for sentences over discrete representation from hand-crafted features for predicting triggers and arguments jointly. Inter-dependencies between triggers and arguments are also captured in memory matrices to store prediction information while labelling sentences (Nguyen, Cho, and Grishman 2016).

In all, like Open IE, event extraction is a distinctive task with more nuance than can be explored further. Grishman has provided a solid base with regards to understanding the literature for event extraction, over Jurafsky and Martin's practical guidance for how it works. From the understanding gleaned from both, however, a decision to disregard this topic when exploring multilingual IE was made for the sake of efficiency, although certain strands were determined to be useful for the experiment.

2.6 Multilingual Information Extraction

At this point, it is worth considering the foundations laid out for IE, as well as returning to the main criticism of Grishman 2019 and D. Jurafsky and Martin 2019a already mentioned. The overviews presented in these comparison sources have given a comprehensive overview but failed to cover methods or literature regarding multilingual IE. Literature in this area is sparse in comparison, but now IE methods have been explored, multilingual methods can be turned to.

At the time ACE was being developed and redefining tasks in IE, the internet was growing in use and size, creating a need for tools to make information from large non-English text collections accessible. At this point, as mentioned above, most non-English IE focused on

[9]Structured perceptron combines a perceptron algorithm for linear classifiers with an inference algorithm, proposed in (Collins 2002)

Japanese. FASTUS (Hobbs et al. 1997) for example was adapted for both English and Japanese and gave comparable results between languages. What is key here though, is that this was not a multilingual solution, and was more a combination of 2 monolingual systems, similar in architecture but with rules tailored to each language. When input consists of multilingual texts, monolingual systems will fail. Other challenges presented when adapting monolingual IE systems into new languages are that users might not know the language of the texts, or that resources such as corpora or linguistic tools might not be available for specific languages (Masche 2004).

The use of terms relating to multilingual systems has also been highlighted as an issue. Its worth solidifying definitions for context here. A monolingual IE system is a system that performs extraction on text input written in a single language that produces results in that same language. A bilingual or multilingual system extracts information from text written in 2 or more defined languages. A language-independent system extracts information from text written in any language. It is also notable that a multilingual system does not need to output results in the same language as it's input text (Masche 2004). This is useful, as we can separate techniques used for language-independent systems and focus on the pipelines of multilingual ones.

There are arguably 2 important aspects of the pipeline that are most apparent compared to monolingual systems. A multilingual system will need an extra step at the beginning to handle language recognition, as this is not language independent. The system would also need to manage the language of extraction results.

The simplest solution for language recognition is to use bigram frequency matrices. The intuition is that each language has distinct conventions such as orthography or word frequency. Proper names and loan words will also affect the accuracy of a language recogniser, which can be alleviated with text heuristics. The best performing models for language recognition are neural networks. For example Y. Jiang, Zhan, and Lan 2020 recently proposed an approach using n-grams combined with word2vec and CNNs, citing accuracy rates as high as 98.61%.

An important consideration when implementing something like language recognition at the start of an IE pipeline is that the components that come next such as a POS tagger or NER tagger will need to process the language of the input text. To do this the components must be adapted to this input. A simpler solution is to treat IE as a black box and utilise MT to translate the input text into a language (usually English) that can be processed by the rest of a monolingual pipeline. Notably this process also regularly relies on language recognition, so this step can be covered by the inclusion of MT. As well as this, the extracted information could be translated into a language appropriate for the user, so the other end of the pipeline could also be managed this way. Figure 5 demonstrates how this looks.

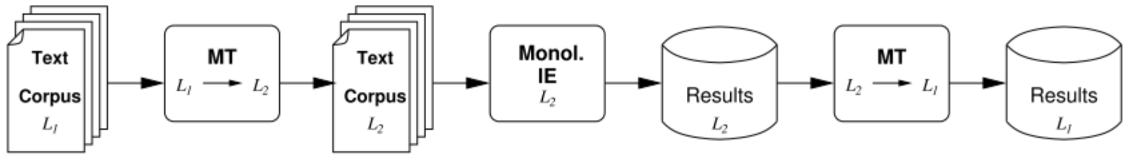


Figure 5: Example of monolingual IE using MT taken from (Masche 2004).

During the period that ACE was introduced, statistical MT was error-prone and ambiguous, but the main benefit advocated for using it in the context of IE was that adapting pipelines for multilingual IE was difficult due to lack of linguistic resources (Masche 2004). The drawbacks reported, however, were more than the benefits. Free text was especially difficult for MT models to handle due to things such as pragmatics, discourse, and idiomatic phrases. Another weakness is that although resources for certain languages may have been lacking for IE, this was also the case for MT. There was a heavy emphasis on developing resources for the big European languages which meant quality was lower for less spoken languages(Masche 2004). Since the resource problem is less prevalent with MT, however, it makes sense to consider how best to utilise that first. The conclusion drawn, however, was that as long as MT performed badly on free text, it was not usable for IE. MT within the IE pipeline has been revisited occasionally in the literature with this in mind.

Faruqui and Kumar 2015 tested a pipeline involving extracting relation tuples from machine-translated sentences and projecting them back onto the source text. A Wikipedia corpus was used and the method involved is as follows:

1. taking sentence $s =< s_1, s_2, ...s_N >$, translating s into English using the Google Translate API to generate $t =< t_1, t_2, ...t_M >$ and saving word alignments relative to s.
2. Performing monolingual relation extraction on t
3. Using word-to-word alignments provided by the Google Translate API to project extracted relation tuples as labels onto s

Rather than using only naive word mapping, an algorithm that incorporates a BLEU[10] score to determine parallel phrases was also used for increased confidence in the projection of the arg1, relation, arg2 tuples. An interesting finding from the study was that some languages such as Russian have a low precision score for extraction even with a high BLEU score for translation. In this case 63.5% precision for 0.62 BLEU. French on the other hand was reported to have 81.6% precision for 0.47 BLEU. Also as expected, the highest number of extracted relations came from bins of phrases with the highest BLEU scores (Faruqui and Kumar 2015). This attests to the original assertion that only higher quality MT would be viable for IE. More importantly, it seems to demonstrate that this method may not be feasible depending on the language. It is also worth reiterating the criticism that this method does not solve the problem of cross-lingual projection to low resource languages.

A recent study seeking to address this problem proposes an approach based on bilingual word embedding mapping (Ni and Florian 2019). There are different means of creating multilingual word embeddings. The first is to create monolingual word embeddings, where vocabulary is mapped to vectors of numbers in a space with many dimensions, normally much smaller than the vocabulary. These embeddings can then be mapped to this same vector space in a different language using a bilingual dictionary. This is the method opted for here, as the other method involves building multilingual word embeddings in a shared vector space simultaneously by using mixed-language corpora (Ni and Florian 2019). Bilingual dictionaries are easier to obtain and thus meet the criteria for minimal resources. This means target embeddings can be found for target text, and cross-lingual mapping gives the English embeddings which are then used as input for a neural network based relation extraction model trained in English.

The scores presented by this approach, however, demonstrate that although it solves the resource problem, it does so with a trade off in performance. The ensemble model combining bi-LSTM and CNN for example shows F_1 scores of 52.4% for Spanish at the highest and 30.2% for Japanese at the lowest (Ni and Florian 2019). So MT still seems to be the most viable option, especially concerning a language like Japanese which may not map well to a similar vector space as English for multilingual embeddings.

[10]BLEU (bilingual evaluation understudy) is an algorithm for determining the quality of machine-translated text

2.7 Summary and Experiment Rationale

This section presented an exploration of the literature surrounding IE through a comparison of 2 comprehensive sources on the topic, Grishman 2019, and D. Jurafsky and Martin 2019a. Through this comparison, that brought in relevant sources to expound certain points and provide examples, the three main areas of IE were explored, NER, relation extraction, and event extraction. The main criticism of the comparison sources is that despite them being overviews, the topic of multilingual IE was not explored. Upon further investigation, it became apparent that the literature in this area was sparse. Nevertheless, some key studies were shown.

The main findings here were that IE systems and their components have been mainly concentrated around extracting from English text, and that linguistic resources for other languages are lacking. Thus, methods for adapting monolingual IE systems trained in English for use in other languages were explored. The key technology here was the utilisation of MT. Whether it was translating free text, translating only extracted results, or projecting results trained in one language to another.

Since the results of multilingual word embeddings were lacking, MT appears to remain the most viable technology for cross-lingual projection, despite its pitfalls. However, disagreement with this sentiment can still be found.

“multilingual methods identified in the literature are still overly dependent on machine translation technology, which we believe is not yet robust enough to be applied” (Claro et al. 2019).

The main question raised then is: is this still the case? Is the quality of MT still low enough as to negatively impact IE performance, especially with the use of neural MT? Looking at the literature there have been some promising results (Faruqui and Kumar 2015), but seemingly only for languages similar to English. It may also be that as with multilingual embeddings, performance for Japanese might be lacking with regards to extracting from it after using MT to translate it into English. As I am familiar with Japanese, this forms a basis for testing, and the experiment presented in this paper. It is also notable that the criticism of MT found in the literature has only focused on MT quality. I believe it would be useful to explore other aspects that contribute to the viability of MT use in the IE pipeline.

3 Methodology

Firstly, due to time and labour constraints, I decided that the experiment to test the viability of MT in the IE pipeline would be kept at a proof of concept level to form a foundation for further research. With this in mind, it was important to focus on a specific IE task, as the literature has presented myriad paths one could branch into in this area. Event extraction for example is a multilayered task usually solved in a modular way, as described in section 2.5. This was disregarded for this experiment, as I wanted a clear focus on the machine-translated text without being concerned about fine tuning an IE task.

Open IE would have been an option to focus on, as from a multilingual IE perspective, this is the most common task in the literature. As shown in section 2.4 however, this was also disregarded. Due to the scope of the project, data collection and system evaluation may have been too onerous, especially when more straightforward options were available as a means of testing IE on machine-translated text.

It was, for this reason, I decided to focus on relation extraction using distant supervision, in a similar way that was used by Mintz et al. 2009, as presented in section 2.4. As demonstrated, building a Wikipedia corpus is relatively quick due to the open-source tools available to extract articles. Mintz et al. 2009 were able to obtain 1.2 million articles for example. Wikipedia also has a wide coverage of information so domain selection is not a problem, and keeping within a specific domain would remove this variable from the experiment. The literature has demonstrated consistently that IE, like translation, performs better when trained within domain confines. Further, Wikipedia articles are written in a consistently structured way that would suit both translation and IE better than noisy user-generated data such as Twitter posts. Using Wikipedia articles is also scalable. As discussed, there is usually a shortage of non-English content available for use in NLP, but Wikipedia provides access to companion articles describing the same entity or concept in various languages. This means the experiment could expand if more languages were to be tested. Finally, with regards to distant supervision, many databases exist that have been created using information extracted from Wikipedia. Using one of these databases to label a Wikipedia article corpus via its attribute extractions would alleviate the necessity of hand annotation. As Freebase was used in the cited Mintz et al. 2009, I decided to use its successor Wikidata. The creators of Wikidata have provided an API by which to query the database intuitively.

3.1 Data Collection

To collect texts within the frame of a specific domain, a useful study to start with was the SHINRA project (Sekine, Kobayashi, and Nakayama 2018) presented in section 2.3. The resources made available by the team were useful concerning this project, as the purpose of SHINRA is to build a Wikipedia based information database like Wikidata in a more structured way. The project is currently in its second year and has already run its 2 tasks successfully on Japanese Wikipedia. The first task was to categorise all NEs found in Wikipedia using Sekine's Extended Named Entities definitions, the second task was to structure all NEs with predefined attributes for each category.

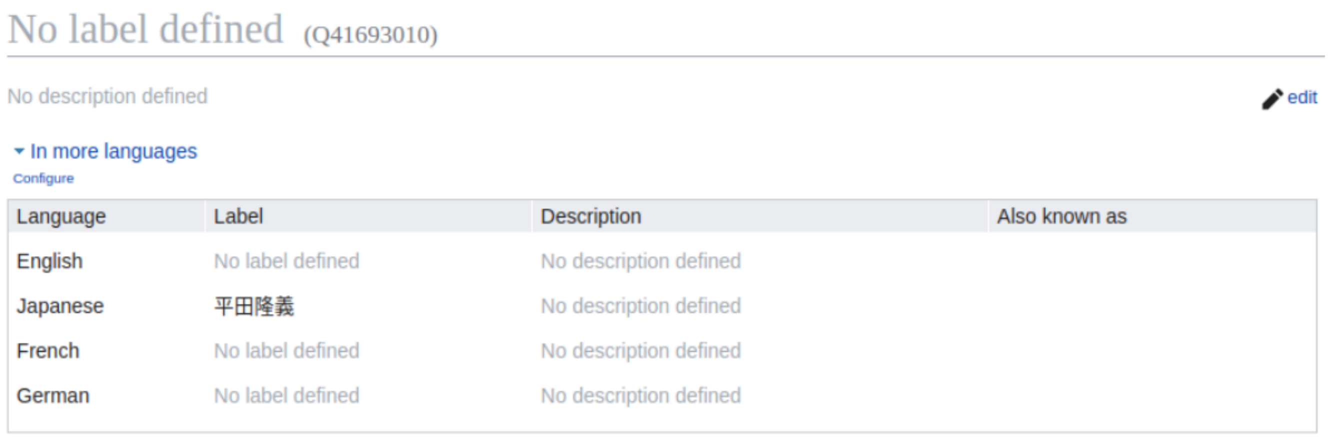
Based on this, the team made available a Japanese Wikidump with the articles based on the project's defined categories. A Wikidump provides a collection of Wikipedia articles listed on Wikipedia in a specific language and for a specified date. This can be performed with open-source Python modules and usually extracts articles, including image placement, infoboxes, reference links, and text body in an XML file. This data is available from the team's website (SHINRA-Project 2020). I opted to download the test dataset containing 401168 articles taken from Japanese Wikipedia and classified into 5 broad categories: Airport, City, Company, Compound, and Person. This dataset would normally be considered small for testing IE, but this size would cut processing time down and still allow for a fair experiment to be conducted. The articles were also extracted with their unique IDs available alongside the article title in the metadata contained within each article text file.

I decided to not utilise the included annotations for entity attributes as these were in Japanese and would have required using cross-lingual projection methods as in Faruqui and Kumar 2015. This would have been an extra layer that added a variable to the experiment that may have detracted from exploration around the main question. The downside is that many articles included in the Wikidump had no English equivalents, and only listed articles about entities exclusive to Japanese Wikipedia. This in itself is interesting, as it demonstrates that this multilingual method could be used to extract new information unseen in English, however, as I chose to use Wikidata to provide my annotation in English, I then had to sort through which articles were Japanese exclusive and which had English equivalents so that I could use articles where information was available. The following table demonstrates the issue I faced when trying to query information for Japanese exclusive articles.

Wikidata ID	Name	Date of Birth	Date of Death	Occupation
Q1155932	Q11559325	1847-01-01T00:00:00Z	1895-01-01T00:00:00Z	none
Q11578602	Hiroshi Hatanaka	1974-01-09T00:00:00Z	none	actor

Table 6: Example Wikidata information query for Japanese exclusive articles.

In the first row, the name label contains a Q number which are unique label IDs used by Wikidata. In this instance, this is due to a label only being defined in Japanese as shown in the figure below.


No label defined (Q41693010)

No description defined

edit

In more languages

Configure

Language	Label	Description	Also known as
English	No label defined	No description defined	
Japanese	平田隆義	No description defined	
French	No label defined	No description defined	
German	No label defined	No description defined	

Figure 6: Example of Wikidata labels being defined in various languages.

The specifics of querying Wikidata will be demonstrated, but for now, the problem of stripping the dataset of Japanese exclusive articles will be shown. Conveniently, the latest task for the SHINRA project currently occurring is to categorise Wikipedia entities for articles in languages other than Japanese. Unfortunately, as this task is not complete, there is no annotation available that I could incorporate into my experiment. However, the team has made available a JSON file containing page ID information to equivalent articles in other languages. I downloaded the links from Japanese to English from their website (SHINRA-Project 2020). A sample line from the file is provided:

```
{"source":{"pageid":33265,"lang":"en","title":"Winston Churchill"},
"destination":{"lang":"ja","title":"ウィンストンチャチル","pageid":41446}}
```

Using this source, extracting Japanese articles with English equivalents was as simple as writing some code to iterate through the Wikidump, taking the page ID of each article and checking if it was contained in the 'language links' JSON file. A cursory check through the articles set aside showed that they all contained no defined labels for English in Wikidata and that they were mostly from the person category, describing historical figures with little text written, especially pages about military figures of medieval Japan. This may hold merit in using IE for historical research, but the relatively short articles would have been lacking in use for model training.

Next, I considered what kind of information I wanted to extract. As discussed in section 2.3, this was necessary for keeping the experiment as simple as possible and keeping domain specificity in mind. By querying every attribute available for each entity in the Wikidump, there was the risk of veering towards optimising for extraction results rather than keeping the focus on testing the machine-translated text. I decided to limit the testing to articles found within the 'person' category as this was the largest batch of articles at 307,641. As well as this, this category seemed like it would contain aspects that would be interesting to test. For example one of the mentioned hypotheses was due to the idiosyncrasies of how Japanese names are written, they would be difficult to translate. 'Person' articles also contain the 'date of birth' and 'date of death' attributes, and as dates are written differently in Japanese compared to English, these seemed interesting to explore. Results and analysis of findings from testing these ideas are described in section 5.2.

The final step for data collection was to extract the text from the remaining articles. Since this had been mostly completed by the SHINRA team, all that was left to do was remove meta information from the beginning and end of the text files. This was as simple as compiling Python code to write all lines, not including lines that started with meta-information noting characteristics that I had identified, to new text files. While writing, I also stripped lines of unnecessary whitespace that remained.

3.2 Translation

After preparing the corpus, the next step was to translate each article from Japanese to English. I began by testing the viability of training my own translation model. The initial plan was to use OpenNMT (Klein et al. 2017); an open-source ecosystem for neural MT. This meant I could focus on testing the machine-translated text with IE its self, rather than dedicating time to the translation task specifically. With this in mind, however, creating a parallel corpus would have been time-consuming. I had a corpus of Japanese articles, and the 'language links' JSON file that I could use to find their equivalents in English. However, Wikipedia articles are not translated across languages. They contain different information depending on the language, and so they can not be used to create parallel corpora easily. This would have also led to overfitting and made the experiment invalid. There was the option of ready-made parallel corpora such as an OPUS parallel corpus (Tiedemann 2012), or using aligned sentences from the Tatoeba project (www.tatoeba.org). In all though, as time and computational resources were limited, this idea was abandoned.

On top of this, despite a thorough search, no pre-trained models for Japanese-English for OpenNMT seemed to be available. The website only lists German or Chinese to English at the time of writing, and so another option was considered. Following the trail for models that have been trained on the Japanese-English OPUS parallel corpus, the best choice seemed to be a model made available by the Language Technology Research Group at the University of Helsinki (Tiedemann and Thottingal 2020). In addition, the team has contributed this, amongst many other models to the *Transformers*[11] library created by Hugging Face (Wolf et al. 2019). This seemed to be a promising option and so I installed the library packages to use with Python and performed some test translations.

After the tests, at the time of experimenting, it was apparent that *Transformers* would be unusable for Japanese. A simple sentence was tested and output the following:

```
Input: 今日はいい天気ですね。
Output: This day shall a good heart to a good To a good heart this day.
Human Translation: It's nice weather today isn't it.
```

Unfortunately, this model had been only trained on aligned sentences from the Bible and did not handle text well outside of this domain. At the time of writing up this experiment, however, a new model has been made available trained on a Tatoeba corpus and boasts a BLEU score of 41.7 (https://huggingface.co/Helsinki-NLP/opus-mt-ja-en), which means it should give high-quality translations. If I were to redo this study, this is now the method I would go with. Being able to translate locally using a pre-trained model would have been a boon.

The only option left was an out of the box solution. At first, taking the lead from Faruqui and Kumar 2015, the Google Translate API was tested. It became apparent, however, that while this solved the translation quality problem, it caused a bottleneck for the experiment.

[11]Transformers is a library dedicated to supporting Transformer-based architectures and facilitating the distribution of pre-trained models (Wolf et al. 2019)

The first was the free version of the API was incredibly limiting, only allowing 500,000 characters to be translated, and cutting off if queried too many times. This would have only let me translate 500 articles. At the time, however, an option was presented with Microsoft Azure (https://azure.microsoft.com/) that allowed 1 million characters to be translated for free, along with free credit. As this was equivalent to the Google Translate API in terms of translation quality, but allowing more to be translated, I went with this option.

The second issue causing a bottleneck highlighted by querying both APIs was the time taken to translate each article this way. I have mentioned that time was a limiting factor, so it was concerning that when testing how long translation was taking using the *time* module in Python, the average for each article in the corpus was 12.4 seconds. Even without the confines of the free trial, if I translated the entire corpus it would average out at 372000 seconds (103.3 hours) of processing time. This was not feasible, but in all, the limits of the free trial meant that only about 1500 articles could be translated. Leaving me with 0.005% of my original data. This was not ideal, but considering the experiment is a proof of concept, it was still acceptable for this paper.

Next, I had to choose which articles would remain in the corpus. Since the corpus was so large, I believed just taking a random sample of 1500 articles would be sufficient to give a good spread of information. In hindsight, this was not the optimal way to select data. If I were to redo this phase, I would have taken a larger random sample, and then extracted attributes related to those articles from Wikidata to analyse the spread of information to make a better decision. For example, I ended up with an imbalanced amount of certain occupation labels, which meant it was difficult to ascertain if my models were learning the pattern for a specific occupation label, or generalising for the features of all occupation relation types. As will be shown in section 3.3, I could not start by labelling the entire corpus due to Wikidata constraining the amount one can query. The random sample method also led to giving me several entities without 'date of death' attributes, as their articles were about living people. This meant that there was not enough of this attribute to feasibly test on. In all, if I redid the experiment I would begin by predefining the arguments for certain types of 'people' entities, and query Wikidata for them first, rather than starting with a Wikidump.

Finally, the corpus of 1500 random 'person' articles was translated into English through the Microsoft Azure API in Python for a total processing time of 42210 seconds (11.7 hours). The translated corpus was then saved for processing and annotation.

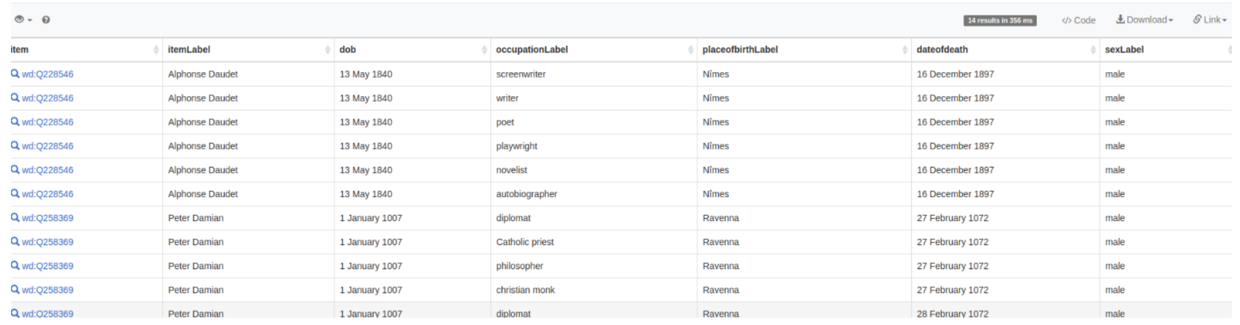
3.3 Wikidata Labels

The next phase was to collect the selected attributes from the Wikidata database for each article in the corpus, to be used for annotation. The Wikidata query service also provides an API that can be used through Python, so this made querying each entity possible by plugging in large variables containing entity query ID numbers for the 'VALUES' arguments taken by the SPARQL[12] query sent to the API. There is an imposed limit on how many arguments can be taken for a single query, so the corpus had to be split into bins of 100 articles each for a total of 15 queries. The API also cuts an IP address making too many requests, but luckily with the now smaller dataset, this was not an issue, but it is worth considering that a different method would be needed should this experiment be scaled up. Example query provided (Where Q ID numbers point to entities and P ID numbers point to attributes):

```
SELECT ?item ?itemLabel ?dob ?occupationLabel ?placeofbirthLabel
?dateofdeath ?sexLabel
WHERE
{
VALUES ?item {wd:Q258369 wd:Q228546}
OPTIONAL {?item wdt:P569 ?dob .}
OPTIONAL {?item wdt:P106 ?occupation .}
OPTIONAL {?item wdt:P19 ?placeofbirth .}
OPTIONAL {?item wdt:P570 ?dateofdeath .}
OPTIONAL {?item wdt:P21 ?sex .}
SERVICE wikibase:label {bd:serviceParam wikibase:language "en".}
}
```

Where there is more than one value extracted per attribute for a single entity, multiple rows are returned as shown in the following example figure, which is the output of the example SPARQL query.

[12]SPARQL is a semantic query language for databases used to retrieve data stored in Resource Description Framework format.



item	itemLabel	dob	occupationLabel	placeofbirthLabel	dateofdeath	sexLabel
wd:Q228546	Alphonse Daudet	13 May 1840	screenwriter	Nîmes	16 December 1897	male
wd:Q228546	Alphonse Daudet	13 May 1840	writer	Nîmes	16 December 1897	male
wd:Q228546	Alphonse Daudet	13 May 1840	poet	Nîmes	16 December 1897	male
wd:Q228546	Alphonse Daudet	13 May 1840	playwright	Nîmes	16 December 1897	male
wd:Q228546	Alphonse Daudet	13 May 1840	novelist	Nîmes	16 December 1897	male
wd:Q228546	Alphonse Daudet	13 May 1840	autobiographer	Nîmes	16 December 1897	male
wd:Q258369	Peter Damian	1 January 1007	diplomat	Ravenna	27 February 1072	male
wd:Q258369	Peter Damian	1 January 1007	Catholic priest	Ravenna	27 February 1072	male
wd:Q258369	Peter Damian	1 January 1007	philosopher	Ravenna	27 February 1072	male
wd:Q258369	Peter Damian	1 January 1007	christian monk	Ravenna	27 February 1072	male
wd:Q258369	Peter Damian	1 January 1007	diplomat	Ravenna	28 February 1072	male

Figure 7: Output of example SPARQL query of Wikidata.

One challenge raised was the retrieval of the Q ID numbers representing entities in the Wikidata database. At this point, I had only collected the Wikipedia article ID numbers. Solving this required multiple steps; the first being to refer again to the 'language links' JSON file provided by the SHINRA project, and then write a Python function that took each article ID in the corpus as input, and iterated through them, matching each ID to the 'destination' property of each object (Japanese article) of the JSON file, and finally returning a list of each 'title' attributes from the 'source' property of each object (English article) where a match was found.

The returned page titles were necessary, rather than page IDs, as these could be used as input for the open-source Python library *wikimapper* (Klie 2020). The module was installed, and the English equivalent article titles of the corpus were saved to an array to be used as input. This essentially was used as a function that took an input title and returned a Q ID number to be used for querying Wikidata, (e.g 'Winston Churchill' to 'Q8016').

Once the Wikidata API had been queried for each article, all attributes were saved into a Pandas data frame object including a column containing article IDs to later annotate the corpus. The multiple rows for each entity, due to multiple occupation attributes, were amalgamated into single rows as shown:

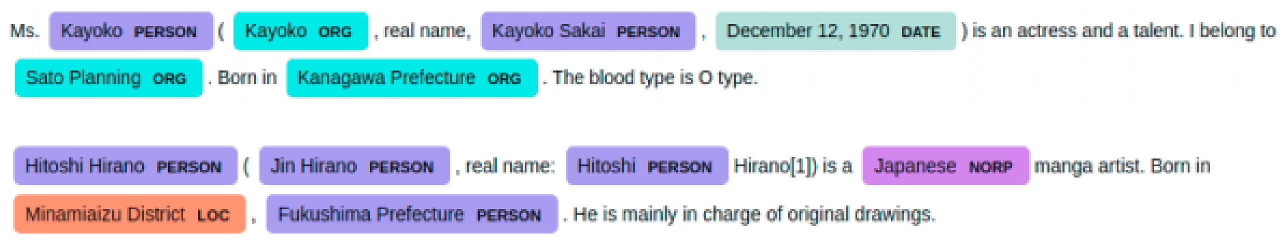
article_id	wikidata_id	article_title	occupation	dob
41446	Q8016	Winston Churchill	historian, politician	1874-11-30T00:00:00Z

Table 7: Sample row of annotation data frame.

3.4 Preprocessing and NER

Before annotation could be done, the text in the dataset had to be NE tagged. This was also the first test of how well the machine-translated text would be handled by the first step in the IE pipeline. SpaCy was selected for its speed and used to tokenize each article by sentence. The sentence was then run through the pipeline to extract POS tags and dependency relations to use for training baseline models selected for the relation extraction phase. The parsed output was saved as XML.

Usefully, the spaCy library also includes an NE tagger so this information was also extracted. Figure 8 shows some example tagged sentences visualised using *displaCy*.


Ms. Kayoko PERSON (Kayoko ORG , real name, Kayoko Sakai PERSON , December 12, 1970 DATE) is an actress and a talent. I belong to Sato Planning ORG . Born in Kanagawa Prefecture ORG . The blood type is O type.

Hitoshi Hirano PERSON (Jin Hirano PERSON , real name: Hitoshi PERSON Hirano[1]) is a Japanese NORP manga artist. Born in Minamiaizu District LOC , Fukushima Prefecture PERSON . He is mainly in charge of original drawings.

Figure 8: Example NE tagged sentences.

As shown, the tagger has picked up each entity in the machine-translated text despite the translations not being accurate syntactically. The NE label types, however, are not perfect, often confusing places and people, For example, ‘Fukushima Prefecture’ has been labelled ‘PERSON’. This was not a problem, however, as the label types were not needed as features. Only the NE positions in the article were used for matching relations in the database for the annotation phase.

The example also shows that dates had been translated and tagged with no problems, so these could be used to find the ‘date of birth’ relations. Because of this, no extra text manipulation was required with regards to dates at this stage of preprocessing. Some extra text cleaning needed was noted such as removing the footnote markers as shown in the second example sentence where only Hitoshi Hirano’s first name was captured.

The main problem as shown in the example was that the tagger could not be used for tagging pronouns or occupations such as ‘actress’. This was solved with string matching, by iterating through each occupation extracted from Wikidata for each entity and then searching each sentence of the matching articles. Pronouns were more tricky as no coreference resolution was attempted for this experiment. Usefully this was not too troublesome due to the structure of Wikipedia articles always beginning with the name of the entity the article is about. The pronouns ‘he’, ‘she’ and ‘I’ (Due to the zero pronoun problem in Japanese described in section 5.2, shown in the first example sentence of figure 8.) were searched for in a similar way to occupations.

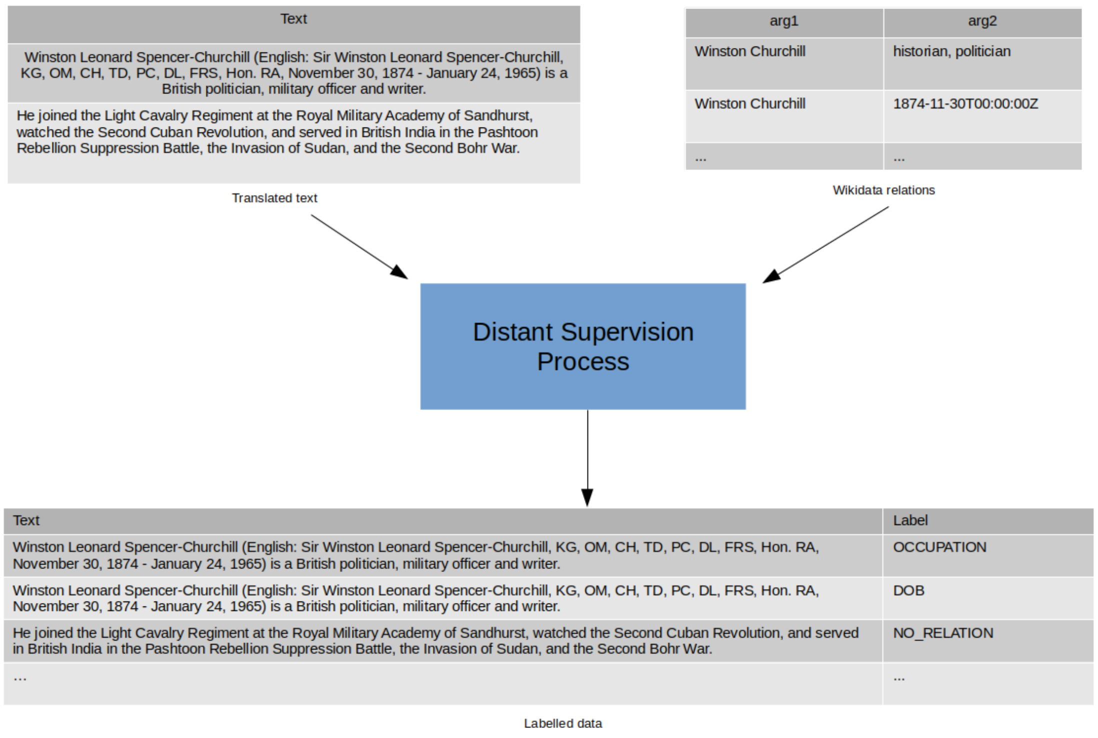
A further issue with occupation tagging, based on the string matching method was the way the translator had handled loanwords. for example, referring back to figure 8, the first sentence shows that the occupation ‘タレント ’ (transliterated as ‘tarento’) had been translated to ‘talent’, whereas the translation in this context is ‘celebrity’. The second example sentence has correctly translated ‘漫画家 ’ (transliterated as ‘mangaka’) to ‘manga artist’. The issue is that ‘mangaka’ is a loanword used in English and so this label was used on Wikidata. To solve this problem I created an array of all occupations, removed duplicates, and translated loanword occupations in the list by hand. This was only viable due to my small dataset (401 occupations) being used at a proof of concept level. A better option may have been to have taken the Japanese labels from Wikidata, rather than the English, and run those through the same translator used on the text. List of translations completed as follows:

mangaka (漫画家) - manga artist, doujin artist
seiyu (声優)- voice actor
owarai tarento (お笑いタレント lit laughing talent) - comedian
tarento - (lit Talent タレント) - celebrity
rakugoka - (落語家) - storyteller (A form of verbal entertainment)
rikishi - (力士) - sumo wrestler
bushi - (武士) - samurai
kabuki actor (歌舞伎) - actor (Kabuki is a type of Japanese dance-drama)
judoka (柔道家) - judo practitioner
tanka poet (短歌) - poet (A type of Japanese short poetry)
aikidoka - (合気道家) - aikido practitioner

Once translations were complete, a Python dictionary was compiled to append the new translations to the occupations column of the annotation data frame. These were then used to find occupations that had slipped through the first time, and the entities were also marked in the tagged sentence data frame. With that, the relation arguments for each article in both data frames were ready to be checked for matches.

3.5 Annotation

Following the distant supervision method (Mintz et al. 2009), to annotate the data I first iterated through each sentence in the sentence data frame, and checked for matches of relevant arg1 and arg2 occurrences (arg1 and arg2 having been taken from the Wikidata data frame). Using the intuition, based on the structure of the articles, that names and pronouns appear as the first argument of a relation if arg1 ('name') from the annotation data frame matched the 'name' entity in the sentence data frame, and the 'OCCUPATION' or 'DOB' labels matched an entity after arg1 in the same sentence, the sentence would be marked with either the relation type found or 'NO_RELATION'. This also follows the intuition that if these entities are found close to each other they will likely express a relation. Matches were maximised by normalising the sentences with temporary lower casing. A conceptual diagram of the process is provided:



Text
Winston Leonard Spencer-Churchill (English: Sir Winston Leonard Spencer-Churchill, KG, OM, CH, TD, PC, DL, FRS, Hon. RA, November 30, 1874 - January 24, 1965) is a British politician, military officer and writer.
He joined the Light Cavalry Regiment at the Royal Military Academy of Sandhurst, watched the Second Cuban Revolution, and served in British India in the Pashtoon Rebellion Suppression Battle, the Invasion of Sudan, and the Second Bohr War.

Translated text

arg1	arg2
Winston Churchill	historian, politician
Winston Churchill	1874-11-30T00:00:00Z
...	...

Wikidata relations

Text	Label
Winston Leonard Spencer-Churchill (English: Sir Winston Leonard Spencer-Churchill, KG, OM, CH, TD, PC, DL, FRS, Hon. RA, November 30, 1874 - January 24, 1965) is a British politician, military officer and writer.	OCCUPATION
Winston Leonard Spencer-Churchill (English: Sir Winston Leonard Spencer-Churchill, KG, OM, CH, TD, PC, DL, FRS, Hon. RA, November 30, 1874 - January 24, 1965) is a British politician, military officer and writer.	DOB
He joined the Light Cavalry Regiment at the Royal Military Academy of Sandhurst, watched the Second Cuban Revolution, and served in British India in the Pashtoon Rebellion Suppression Battle, the Invasion of Sudan, and the Second Bohr War.	NO_RELATION
...	...

Labelled data

Figure 9: Distant supervision process.

As the diagram indicates, the text and labels were combined together in a new single data frame ready to be used for model training. As expected however, the process of distant supervision showed false positives and false negatives at the this stage, especially in conjunction with low quality translation. An example of a false negative is taken from the data frame:

id	arg1	arg2	relation	text
1003857	Nils Katayainen	military personnel	NO_RELATION	It is generally known...

Table 8: Sample line containing false negative annotation.

```
text: It is generally known by the name of [arg1 Katayainen] which does
not follow because it was an ace [arg2 pilot] who raised a lot of war...
```

Table 8 shows that although the the first argument was partially matched, the label taken from Wikidata 'military personnel' did not match the second argument 'pilot', and it was erroneously marked as 'NO_RELATION'. This is a weakness of the distant supervision method that is usually overcome by having a larger dataset. For this experiment, no corrections were made, which in hindsight may have added too much noise to the dataset. An example of a false positive is also given:

id	arg1	arg2	relation	text
976	Hayami Tsubasa	mangaka, artist	OCCUPATION	Tsubasa had been dating...

Table 9: Sample line containing false positive annotation.

```
text: [arg1 Tsubasa] had been dating manga [arg2 artist] Shuichi Shige
for many years and married for a while.
```

This also shows a weakness in only looking for certain relations, as opposed to trying to find all relations using a different method. For example, even though 'Tsubasa' and 'Shuichi Shige' were labelled as entities, there may have been no attribute in Wikidata denoting a relation between them, despite one existing. This is where open IE would be a preferred method.

After all the sentences had been labelled, some further processing was completed. First, any sentences shorter than 5 tokens in length, and marked with 'NO_RELATION' were taken out of the dataset. This still left an uneven ratio of no relation to relation labelled sentences, so a function was written utilising the sample method of a Pandas data frame to return a sample of random sentences marked 'NO_RELATION' equal to the amount of relation marked sentences in the dataset. This left the final dataset with 5662 labelled sentences, which was a reasonable amount to begin testing models with, and still more than could have been achieved with a purely supervised method of annotation given the time limitations.

4 Results

After completing the labelled data frame, I made several adjustments based on the selected models that were to be trained. I decided on a simple bag-of-words approach with trigrams to maintain word order as the first method to be used to discover a baseline result. The sentences were first lowercased, as this captured more trigrams. Next, I imported the *Scikit Learn* library and used the 'count vectorizer' class to take each row of text from the data frame as input, and output the text trigrams, represented as numerical vectors. These were then saved into a feature matrix. After this, I converted labels where a relation was expressed as a positive label (1), and all 'NO_RELATION' labels were expressed as negative (0). Based on the literature, this would test a filter layer that would determine if a sentence expressed a desired relation or not.

The second layer involved removing all 'NO_RELATION' sentences and only keeping the sentences that expressed relations. After doing this, the same process was repeated to create a new feature matrix. The label array this time, however, involved converting the 'OCCUPATION' labels to 1, and the 'DOB' labels to 0. The earlier decision to only focus on 2 relations for the sake of proof of concept allowed me to keep this level as a straightforward binary classification task.

The feature matrices and label arrays for both layers were then tested on a linear support vector classifier and a logistic regression classifier using 5 fold cross validation. Results are displayed in the following tables:

Layer	Accuracy	Precision	Recall	F_1
Relevance Classification	0.743	0.863	0.587	0.676
Relation Classification	0.678	0.665	0.717	0.69

Table 10: Baseline results using trigrams for SVC model.

Layer	Accuracy	Precision	Recall	F_1
Relevance Classification	0.744	0.868	0.581	0.672
Relation Classification	0.676	0.665	0.71	0.686

Table 11: Baseline results using trigrams for logistic regression model.

After obtaining a baseline, I decided to bring in the syntactic features that were previously extracted from the raw text of the labelled data frame using spaCy. These features were added to the feature matrices and new models were trained. Results for both classifying relevance of sentences and relation type classification were again obtained using the new features. These are included in tables 12 and 13.

Layer	Accuracy	Precision	Recall	F_1
Relevance Classification	0.725	0.786	0.628	0.683
Relation Classification	0.696	0.692	0.707	0.7

Table 12: Results using trigrams and syntactic features for SVC model.

Layer	Accuracy	Precision	Recall	F_1
Relevance Classification	0.731	0.799	0.626	0.684
Relation Classification	0.692	0.688	0.701	0.694

Table 13: Results using trigrams and syntactic features for logistic regression model.

Finally, despite not having a large dataset, I wanted to test deep learning methods, as shown in much of the latest literature. For this I opted to refer back to the Transformers library (Wolf et al. 2019), and selected the RoBERTa[13] model. This model was used through the *Simple Transformers* library (Rajapakse 2020), and as no feature engineering was needed, only the raw text along with labels for both relevance and relation classification saved into data frames were used as input. Again, as the focus was on testing the viability of machine-translated text, rather than optimising extraction, I opted to keep the default hyperparameters. Results were a big improvement over the statistical models as shown below:

[13]RoBERTa builds on the language masking strategy of BERT through modified hyperparameters, training with larger mini-batches and learning rates, etc (Y. Liu et al. 2019)

Layer	Accuracy	Precision	Recall	F_1
Relevance Classification	0.844	0.827	0.87	0.848
Relation Classification	0.802	0.91	0.669	0.771

Table 14: Results obtained from RoBERTa model.

5 Discussion

5.1 Examination of Results

Firstly, while observing baseline results, the difference between SVC and logistic regression are negligible, although looking purely at F_1, SVC is slightly better. The results at this stage are promising in terms of the utilisation of machine-translated Japanese text in an IE pipeline, as they are comparable with the results from other studies. In terms of distant supervision Mintz et al. 2009 reported 67.7% precision using English Wikipedia with English trained statistical models and feature extraction. However many relations were captured, which differs from the experiment presented here.

Results for relevance classification using the statistical models, regardless of whether syntactic features were used to bolster performance or not, shows there is a problem of high precision, low recall. This means that at the level of detecting if text contains a relevant predefined relation, the models were relatively accurate when they pick out sentences to be labelled, but the lower recall means that many sentences that contain relevant relations were not picked up. It is difficult to determine if this was due to machine-translated text being used, or just the lack of features, as recall and precision were balanced out more when syntactic features are introduced. These results are still useful, however, as they show a likely issue with the distant supervision method in combination with the dataset its self. As shown the distant supervision method tends to end up with false positive and false negative labels. As well as this, Wikipedia articles are structured in similar ways so that when only looking for predefined relations, only sentences similar in structure may have been labelled. These combining factors may have led to the classifiers missing the fewer in number relation expressed sentences due to being incorrectly trained to label other types of sentences as relevant, due to false positives appearing in the dataset. The problem of catching more relevant sentences seems to be resolved by using the RoBERTa model which seems to show the lack of features hypothesis may have been correct, as the arbitrary features picked up from the data by the model has increased recall to 87%. Based on the lower precision though, it seems to be classifying non-relevant sentences as relevant. This may be again that Wikipedia articles are written in similarly structured ways and that I was only looking for specific relations, rather than all relations. The sentences that contained relevant relations that I was not searching for may have been picked up, as they were similar to sentences that contained relations I was searching for. Due to the size of the dataset used, however, that hypothesis must remain conjecture for now. For future work, resolving the bottleneck at the translation stage would allow the dataset to be kept large enough for more effective conclusions to be drawn from neural network results.

With regards to the second layer of classifying the relation type when a relevant sentence is found, the baseline models seem balanced, but the high recall to lower precision indicates the dataset might have been too imbalanced. Due to the structure of Wikipedia articles, there were more sentences expressing the 'OCCUPATION' relation than there were sentences expressing the 'DOB' relation, which is normally only mentioned once in the article. This means that when classifying 'DOB', it was a minority class, making the dataset contain a lot of negative sentences that became false positives. This problem was balanced out when introducing syntactic features.

When looking at the RoBERTa model, relation classification showed the opposite results. Precision was 91% but recall was as low as 66.9%. After looking through the dataset, one explanation could be that many sentences had conflicting labels due to the distant supervision process, as shown in figure 9. This introduced noise into an already limited dataset that was skewed towards 'OCCUPATION' labels. In this case, it stands to reason that for classifying relation types, the RoBERTa results demonstrate the noise introduced caused too much loss for 'DOB' labels and therefore the model did not think many 'DOB' labelled sentences contained 'DOB' relations. The learning point from this is that more relations should have been searched for and included to provide a balance in the dataset, although at the risk of introducing more mislabelled samples during the distant supervision process.

There is also the concern raised from the relatively high F_1 scores for both layers provided by the RoBERTa model, despite the dataset being proven to be imperfect. In this instance, there is suspicion of the model overfitting. Overall then, it may be that this experiment was not thorough enough in terms of model results to demonstrate if using MT in the IE pipeline is viable. However, the problems raised do not seem to stem directly from the quality of the translation used and at least in the case of detecting relations at the relevance level, results for the statistical models are similar to if natural English articles were used. One means of testing this would have been to run English articles through the same pipeline, minus the translation, and compare results. Had time permitted, I would have done this. As it stands, however, the answer is inconclusive, so as explained in the experiment rationale, other aspects were explored.

5.2 Further Analysis

As stated, while Wikipedia articles are written in such a way that may have led to overfitting, the benefits of using them outweighed the negatives. If the dataset could have remained its original size, and the task was switched to open IE, this may have proven to be a better test and minimised any potential overfitting issues. In other words, the original dataset was fine, it was the factors introduced further into the pipeline that caused issues.

The first solution for keeping the original dataset size has already been mentioned, that switching to a local translation model would overcome the imposed monetary limits of using cloud-based translation, but this does not address the issue of time. As demonstrated, taking over 11 hours to translate only 1500 articles using Microsoft Azure was another bottleneck. Its difficult to say if local translation would improve these times, and when a system is trained this will not be as much of an issue, but when compared to using a monolingual system and translating the results, in terms of time, using MT in this way is leaning towards being infeasible as the literature claims.

Mentions of the weakness of distant supervision have also been made throughout the paper, and there is some evidence that low-quality translation does not mesh well with this method. These issues also show up at the NER stage. I have already explored the problem of how occupations were translated compared to the labels available for them on Wikidata, and that a solution might have been to translate the Japanese Wikidata labels as opposed to using the English labels. This solution seems less viable after examining the 'NAME' arguments found by NER and comparing them to what was extracted from Wikidata, which as mentioned in section 3.1 is something I had wanted to test. It was apparent that potential matches were missed due to the difficulties of how names are read in Japanese, and many were only picked up due to partial matches. In Japanese, individuals with Japanese names usually have names written as a compound of kanji[14] characters. In usual circumstances,

[14]The Japanese writing system is a combination of kanji, which are adopted logographic characters from China, literally meaning "Han characters", and two syllabic scripts called hiragana and katakana.

kanji compounds are usually read by their 音読み(onyomi)[15] reading, which is the Chinese reading of the kanji. Names on the other hand may also use kanji that can be read using the 訓読み(kunyomi)[16], which are the more numerous Japanese readings. In other words, a name can be read a myriad of ways and can even trouble native speakers. An example in the dataset is 広瀬佳一 (Yoshikazu Hirose), which was translated both as 'Keiichi Hirose' and 'Hirose Yoshikazu' in the article. Family names come first in the compound, but it was chance if the positions of the family and given names were retained, or translated correctly.

This problem also extends to how Chinese names are translated. Since kanji are Chinese characters, they also appear in Chinese names, which are written the same way in Japanese. This is problematic if the translation model uses a Japanese reading, as opposed to the correct Chinese reading for a character. An example of this occurred while translating the sample text of the entity 胡钁(Jue Hu). This was translated to 'Ebisu钁', which is interesting for a few reasons. First '胡' can be read as 'Ebisu', which is a Japanese family name. Once transliterated, it can lose its meaning as a name and could be labelled as a place at the NER stage. Secondly, '钁' was not translated at all. In Japanese, this character is so obscure that to my knowledge it is not used. This demonstrates another weakness of using MT in the IE pipeline.

Names written in other writing systems were also problematic at the translation stage. In Japanese, names that cannot be written with kanji or hiragana, as well as loanwords, are written in katakana, a syllabary used to represent non-Japanese words phonetically. The issue is that Japanese contains a total of 22 phonemes, a small amount relative to most other languages, which means using katakana to represent the phonemes of other languages is difficult. This in turn makes katakana difficult to translate for rare words like foreign names, for example, into English where various letters could be used in place of 1 katakana character. An example of where this occurred in the dataset is with the entity 'アレクサンデル・パーブロビッチ・ロバノフ' (Aleksander Pavlovitch Lobanov). This was output as 'Alexander Papuro Bitch Robanofu' which led to a missed label. Since this was translated to English, 'アレクサンデル' was translated to the more commonly used 'Alexander', and 'ロバノフ' to 'Robanofu' is an example of where the translation model used literal phonetic transliteration. It also shows how the character 'ロ' can be represented as 'ro' or 'lo' because the Japanese liquid consonant [ɾ] does not appear at the beginning of words in English, and is often perceived as either /r/ or /l/.

[15]'Onyomi' refers to readings of kanji derived from their Chinese pronunciations.
[16]'Kunyomi' refers to the indigenous Japanese pronunciations of a kanji.

In all, without partial matching, it is uncertain if this experiment would have succeeded. After testing Wikidata labels against the translated names at the beginning of each article using string matching, the accuracy was only 36.4%. Although this problem is specific to Japanese, it demonstrates a flaw to be resolved for the method presented in this experiment and maybe a strong reason for false negatives in the distant supervision process. A Japanese name dictionary could go some way to resolving this, but it is difficult to resolve the problem of literal transliterations of rare katakana names, as monolingual spelling correction techniques focus on cognitive errors or typos (mostly missed or added letters to already existing words). In this instance, a spell checker tailored to learners of English as an L_2 comes to mind, but this would veer towards creating a bilingual IE system, rather than a true multilingual solution.

Another quirk of Japanese, as mentioned previously is the problem of how zero-pronouns are handled at the translation stage. The omission of pronouns when they are understood from context as being grammatically correct in Japanese is problematic when translating, as pronouns need to be inserted. When a name appears in the text, 'he' or 'she' is inserted. If no name appears, a reader would normally assume the writer is self-referring, and so the pronoun 'I' will be inserted by MT, but this is not always the case. Figure 8 gave the example translated phrase:

```
Original: 伽代子（かよこ、本名・坂井 香代子、1970年12月12日 - ）は、女優、タレント。佐藤企画に所属している。神奈川県出身。血液型はO型。

Machine Translation: Ms. Kayoko (Kayoko, real name, Kayoko Sakai,
December 12, 1970 ) is an actress and a talent. I belong to Sato Planning.
The blood type is O type.

Human Translation: Kayoko (real name: Kayoko Sakai, 12/12/1970 - ) is
an actress and celebrity. She belongs to the Sato Kikaku agency.
Her blood type is O.
```

Here, we see the honorific 'Ms' has been inserted, as the nickname (written using different kanji as shown) has been detected as a family name. Interestingly, an honorific used for women has been correctly selected, likely because names ending in '子'('ko') are feminine, and this has been picked up at the training stage of the used model. 'I' rather than 'she' has been erroneously inserted in the second sentence, and ‘The’ has been inserted instead of ‘Her’ in the final sentence, as this looks to have been literally translated.

Until these problems are resolved, NER on machine-translated Japanese text could prove difficult. It would be interesting to see how coreference resolution would fare on the machine-translated text also. If it were performed on the text before translation, the fact there is a lack of entities at all within the text could mean that coreference resolution tailored to Japanese would be required. This is problematic as again, it leans towards a bilingual solution at best. If this experiment could be expanded, this is one of the things I would like to test.

The example does show that dates were handled well and picked up at the NER stage. As mentioned in the experiment rationale, this was something I was concerned about. In the literature review, I had anticipated needing to map dates in the corpus to the ISO 8601 standard, especially as the date labels extracted from Wikidata were in this format. However, the MT model and NER stages handled these and made string matching easy.

In terms of translation quality, even with the issue of pronoun insertion, the text seems viable to extract from, and despite the issues, this is backed up by the results. If the quality of the translation was unusable, then results comparable to monolingual extraction would not have been achieved. When comparing the syntactic features of the machine-translated example compared to the human-translated example, one can see that the main features are the same in both cases. See figures below for example:

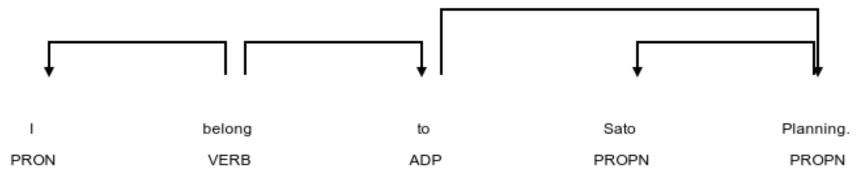


Figure 10: POS and dependency relations of machine-translated sentence.

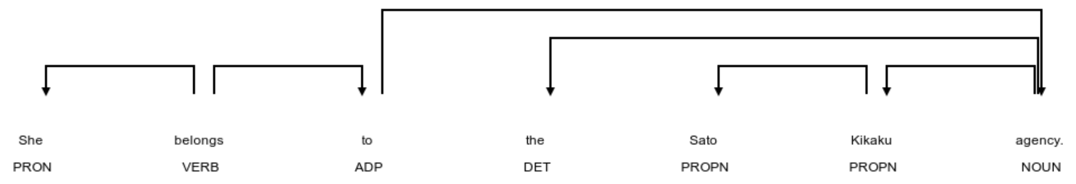


Figure 11: POS and dependency relations of human-translated sentence for comparison.

When the translated sentences are visualised, it is possible to see how simple in structure Japanese is compared to English. Arguably a more simple representation of a sentence would be easier to extract from, and word embeddings seem to eliminate the issue of lack of features. This again, however, is specific to Japanese, and so to expand this experiment to other languages for comparison is the next logical step after solving the issues already brought up.

Some other solutions to improve this experiment, given more time, were also found in the literature. For example Faruqui and Kumar 2015 relied on BLEU scores when deciding what to retain in their training data. As I have only examined translation quality by eye, implementing quality estimation may give better metrics for understanding how extraction is affected by translation quality.

As well as this, a method for improving distant supervision trained models by infusing the training data with a small amount of human labelling has been demonstrated to significantly increase the F-score over pure distant supervision by 13.5% (Pershina et al. 2014). If distant supervision is used again rather than open IE, implementing this method could be beneficial, however, this is not a priority, as this is focused on improving IE results rather than directly testing MT in the pipeline.

6 Conclusion

Overall, when addressing the original questions expressed in the introduction and rationale , the experiment as a whole has revealed a nuanced set of answers. In this paper, I performed relation extraction on machine-translated Japanese Wikipedia articles to test three questions. Is MT viable in the IE pipeline? Can information be extracted from low-quality machine-translated text? How suitable is MT as a multilingual solution to IE?

With regards to the first two questions, the observation of the results are comparable to extraction tasks performed on natural language with a monolingual IE system. This, coupled with the exploration of the features of machine-translated text by itself self suggests that using MT to perform IE is viable, and that low-quality text can be used for extraction. However, when exploring the pipeline in more detail, a set of bottlenecks were made apparent that do challenge these findings somewhat.

Most solutions to problems presented in the analysis only work to solve problems introduced by the idiosyncrasies of Japanese specifically. The original hope that the methods introduced in this paper could be scaled to include other languages may not be possible. Problems that other languages may introduce might stack, or vary in structure from Japanese to English machine-translated text. This could be unusable for training a multilingual system. This lines up with the assertion described in the literature that MT should not be relied on in the framework of multilingual IE and also the third research question presented by this paper. However, the results do indicate that the machine-translated dataset has been extracted from relatively successfully, with minimal manual intervention needed on the translated text. Perhaps if the suggested fixes mentioned throughout the paper are implemented, it is arguable that this experiment has demonstrated using MT in the IE pipeline is promising, and could be feasible for bilingual systems, if not multilingual ones.

The most important of these fixes is arguably the problem of losing the size of the original dataset due to the translation method used. Any conclusions drawn so far are limited because of this, but the hope of completing a proof of concept was still achieved. As well as this, event extraction was not tested due to the scope of the paper. Multilingual event extraction is also lacking in the literature so conceivably there is still a lot of scope left for testing MT for use in other areas of multilingual IE.

Perhaps another shortcoming of the project was the lack of comparison methods. In section 2.6 the possibility of using multilingual word embeddings was addressed, but MT seemed to be a more viable method based on a comparison of the results. I believe there is merit in revisiting this avenue, especially if other languages were also compared, because as shown, the focus on Japanese may have blocked useful information that could have been explored. Also mentioned already, but running natural English articles through the same architecture may have led to drawing further conclusions regarding the research question. These are implications that need to be considered for future work in this area.

Further implications may be that this experiment has given weight to the view that MT is still not robust enough to add to a process that is already complicated by multi-stage issues. Components in IE at each stage tend to introduce errors that may be magnified by each successive stage, and as shown by this experiment, even state of the art MT is error-prone. However, there is certainly promise in MT and it remains the best technology when considering multilingual IE systems. The identified unexplored points brought up in this paper also demonstrate there are good prospects for improvement in this area of research.

References

Ahn, David (2006). “The stages of event extraction”. In: *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pp. 1–8.

Akbik, Alan, Tanja Bergmann, and Roland Vollgraf (2019). “Pooled contextualized embeddings for named entity recognition”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 724–728.

Bollacker, Kurt et al. (2008). “Freebase: a collaboratively created graph database for structuring human knowledge”. In: *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 1247–1250.

Chiticariu, Laura, Yunyao Li, and Frederick Reiss (2013). “Rule-based information extraction is dead! long live rule-based information extraction systems!” In: *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 827–832.

Claro, Daniela Barreiro et al. (2019). “Multilingual open information extraction: challenges and opportunities”. In: *Information* 10.7, p. 228.

Collins, Michael (2002). “Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms”. In: *Proceedings of the 2002 conference on empirical methods in natural language processing (EMNLP 2002)*, pp. 1–8.

Doddington, George R et al. (2004). “The automatic content extraction (ace) program-tasks, data, and evaluation.” In: *Lrec*. Vol. 2. 1. Lisbon, pp. 837–840.

ExplosionAI (2020). *spaCy Annotation Specifications*. URL: `https://spacy.io/api/annotation` (visited on 08/31/2020).

Faruqui, Manaal and Shankar Kumar (2015). “Multilingual open relation extraction using cross-lingual projection”. In: *arXiv preprint arXiv:1503.06450*.

Fellbaum, Christiane (2012). “WordNet”. In: *The encyclopedia of applied linguistics*.

Gillings, Michael R, Martin Hilbert, and Darrell J Kemp (2016). “Information in the biosphere: Biological and digital worlds”. In: *Trends in ecology & evolution* 31.3, pp. 180–189.

Grishman, Ralph (2019). “Twenty-five years of information extraction”. In: *Natural Language Engineering* 25.6, pp. 677–692.

Grishman, Ralph and Beth M Sundheim (1996). “Message understanding conference-6: A brief history”. In: *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Hobbs, Jerry R et al. (1997). “FASTUS: A cascaded finite-state transducer for extracting information from natural-language text”. In: *Finite-state language processing*, pp. 383–406.

ISO8601, ISO (2004). “data elements and interchange formats–information interchange–representation of dates and times”. In: *Geneva: International Organization for Standardization.*

Jiang, Jing and ChengXiang Zhai (Apr. 2007). “A Systematic Exploration of the Feature Space for Relation Extraction”. In: *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*. Rochester, New York: Association for Computational Linguistics, pp. 113–120. URL: `https://www.aclweb.org/anthology/N07-1015`.

Jiang, Yanhong, Xuegang Zhan, and Yixiao Lan (2020). “Multilingual Recognition Based on Neural Network with High Recognition Degree”. In: *Journal of Physics: Conference Series*. Vol. 1549. 5. IOP Publishing, p. 052056.

Jurafsky, Dan and James H Martin (2019a). *Speech & language processing - 18.Information Extraction*. 3rd ed. Unpublished - Available at https://web.stanford.edu/ jurafsky/slp3/.

— (2019b). *Speech & language processing - 22.Coreference Resolution*. 3rd ed. Unpublished - Available at https://web.stanford.edu/ jurafsky/slp3/.

— (2019c). *Speech & language processing - 22.Question Answering*. 3rd ed. Unpublished - Available at https://web.stanford.edu/ jurafsky/slp3/.

Klein, Guillaume et al. (2017). “Opennmt: Open-source toolkit for neural machine translation”. In: *arXiv preprint arXiv:1701.02810*.

Klie, Jan-Christoph (2020). *wikimapper 0.1.5*. URL: `https://pypi.org/project/wikimapper/` (visited on 06/17/2020).

Li, Qi, Heng Ji, and Liang Huang (Aug. 2013). “Joint Event Extraction via Structured Prediction with Global Features”. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 73–82. URL: `https://www.aclweb.org/anthology/P13-1008`.

Liu, Yinhan et al. (2019). “Roberta: A robustly optimized bert pretraining approach”. In: *arXiv preprint arXiv:1907.11692*.

Masche, Philipp (2004). “Multilingual information extraction”. In: Helsingfors universitet.

Mintz, Mike et al. (Aug. 2009). “Distant supervision for relation extraction without labeled data”. 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. Suntec, Singapore: Association for Computational Linguistics, pp. 1003–1011. URL: https://www.aclweb.org/anthology/P09-1113.

Nadeau, David and Satoshi Sekine (2007). "A survey of named entity recognition and classification". In: *Lingvisticae Investigationes* 30.1, pp. 3–26.

Nguyen, Thien Huu, Kyunghyun Cho, and Ralph Grishman (June 2016). "Joint Event Extraction via Recurrent Neural Networks". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, pp. 300–309. DOI: 10.18653/v1/N16-1034. URL: https://www.aclweb.org/anthology/N16-1034.

Ni, Jian and Radu Florian (2019). "Neural Cross-Lingual Relation Extraction Based on Bilingual Word Embedding Mapping". In: *arXiv preprint arXiv:1911.00069*.

Pershina, Maria et al. (2014). "Infusion of labeled data into distant supervision for relation extraction". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 732–738.

Peters, Shanan E et al. (2014). "A machine reading system for assembling synthetic paleontological databases". In: *PLoS one* 9.12, e113523.

Rajapakse, Thilina (2020). *Simple Transformers*. URL: https://github.com/ThilinaRajapakse/simpletransformers (visited on 07/27/2020).

Schank, Roger C and Robert P Abelson (1977). *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures*. Psychology Press.

Sekine, Satoshi and Hitoshi Isahara (2000). "IREX: IR & IE Evaluation Project in Japanese." In: *LREC*. Citeseer, pp. 1977–1980.

Sekine, Satoshi, Akio Kobayashi, and Kouta Nakayama (2018). "SHINRA: Structuring Wikipedia by Collaborative Contribution". In: *Automated Knowledge Base Construction (AKBC)*.

SHINRA-Project (2020). *SHINRA Data Download*. URL: http://shinra-project.info/download/?lang=en (visited on 04/28/2020).

Snow, Rion, Jurafsky, and Andrew Y Ng (2005). "Learning syntactic patterns for automatic hypernym discovery". In: *Advances in neural information processing systems*, pp. 1297–1304.

Sudo, Kiyoshi (July 2001). "Japanese Information Extraction with Automatically Extracted Patterns". In:

Tiedemann, Jörg (2012). "Parallel Data, Tools and Interfaces in OPUS." In: *Lrec*. Vol. 2012, pp. 2214–2218.

Tiedemann, Jörg and Santhosh Thottingal (2020). “OPUS-MT — Building open translation services for the World”. In: *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*. Lisbon, Portugal.

w3techs (2020). *Usage statistics of content languages for websites*. URL: `https://w3techs.com/technologies/overview/content_language` (visited on 06/25/2020).

Wang, Yanshan et al. (2018). “Clinical information extraction applications: a literature review”. In: *Journal of biomedical informatics* 77, pp. 34–49.

Wikipedia (2020). *Wikipedia Winston Churchill*. URL: `https://en.wikipedia.org/wiki/Winston_Churchill` (visited on 08/31/2020).

Wolf, Thomas et al. (2019). “HuggingFace’s Transformers: State-of-the-art Natural Language Processing”. In: *ArXiv* abs/1910.03771.