MULTIMUC: Multilingual Template Filling on MUC-4

Anonymous ACL submission

Abstract

We introduce MULTIMUC, the first multilingual parallel corpus for template filling, comprising translations of the classic MUC-4 template filling benchmark into five languages: Arabic, Chinese, Farsi, Korean, and Russian. We obtain automatic translations from a strong multilingual machine translation system and manually project the original English annotations into each target language. For all languages, we also provide human translations for key portions of the dev and test splits. Finally, we present baselines on MULTIMUC both with state-of-the-art template filling models for MUC-4 and with ChatGPT. We release MULTIMUC and the supervised baselines to facilitate further work on document-level information extraction in multilingual settings.

1 Introduction

The Message Understanding Conferences (MUCs) were a series of U.S. government-sponsored competitions that ran from the late 1980s through the late 1990s whose aim was to promote the development of systems for extracting complex relations from text, and which have been credited with inaugurating the field of information extraction (IE; Grishman and Sundheim, 1996; Grishman, 2019). The third MUC (MUC-3) introduced the now classic task of *template filling*, in which systems must identify events, represented by predefined schemas or *templates*, in a document, and populate roles or *slots* in those templates with relevant information extracted or inferred from the text (muc, 1991). The MUC-3 task focused on identifying various forms of terrorism (e.g. bombings, kidnappings) in news reports from a number of countries in Latin America. Systems had to extract one template per

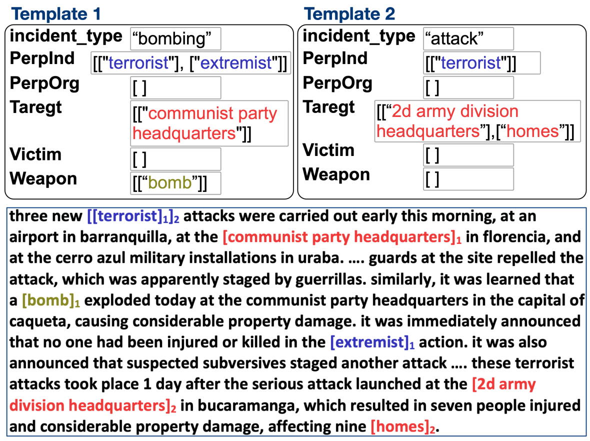

Figure 1: An excerpted document and its (simplified) gold templates from the MUC-4 dataset.

incident, containing details about the perpetrators, their victims, the weapons used, and the infrastructure targeted. The data, task specification, and evaluation methodology of MUC-3 were then refined and updated in MUC-4 (muc, 1992).

Since then, the MUC-4 corpus has been an enduring and productive driver of IE research — not only for template filling (Du et al., 2021b; Das et al., 2022; Chen et al., 2023b) and role-filler entity extraction (Patwardhan and Riloff, 2007, 2009; Huang et al., 2021; Du et al., 2021a), but also for template *induction* (Chambers and Jurafsky, 2011; Cheung et al., 2013). But despite its multinational focus, MUC-4 is English-only, and multilingual, document-level IE datasets remain scarce. This work bolsters those resources with **MULTIMUC**, the first ever translations of the MUC-4 dataset, and to our knowledge the first multilingual *parallel* corpus for template filling. This work provides:

- High-quality, automatic translations of the MUC-4 dataset into five languages: Arabic, Chinese, Farsi, Korean, and Russian, along with (1) manual projections of the template annotations into each target language, and (2) expert human translations for key portions of the dev and test splits.

- Strong monolingual and bilingual supervised baselines for all five languages, based on state-of-the-art template filling models.
- Baselines for few-shot template filling with ChatGPT[1] — to our knowledge, the first few-shot evaluations of this task in the literature.
- Discussion and analysis of the translations, annotations, and model errors.

All data, as well as our MT system and supervised baselines, will be made publicly available to help further research in multilingual, document-level IE.

2 Task and Corpus

Task Formally, the template filling task takes the following inputs:

- A document $D = (w_1, \ldots, w_L)$, consisting of words w_1 to w_L
- A template ontology $(\mathcal{T}, \mathcal{S})$, consisting of a set of template types $\mathcal{T} = \{T_1, ..., T_M\}$, each representing a distinct event type, as well as a set of N_t slots for each template type $t \in \mathcal{T}$, representing the roles for that event type: $\mathcal{S} = \{S_t = \{s_t^{(1)}, \ldots, s_t^{(N_t)}\} : t \in \mathcal{T}\}$

Given D, systems must then determine the number of events or *template instances* ($N_D \geq 0$) attested in D (**template identification**), and populate the slots in each instance based on the information contained in D about the event it represents (**slot filling**).[2] Note that N_D is not given as input and may be zero; thus, part of the task is determining the *relevancy* of a document given the ontology. Supposing instance $i_j \in \{i_1, \ldots, i_{N_D}\}$ has type $t \in \mathcal{T}$, we can write $i_j = \{s_t^{(1)} : x^{(1)}, \ldots, s_t^{(N_t)} : x^{(N_t)}\}$, where $x^{(k)}$ is a (possibly null) filler of the appropriate type for slot $s_t^{(k)}$. In general, fillers may be of any type, though for MUC-4, they are constrained to two types in principle and just one in practice (see below).

Corpus The MUC-4 corpus consists of 1,700 documents that broadly concern incidents of terrorism and political violence in Latin America and that are annotated against a template ontology with six template types: `arson`, `attack`, `bombing`, `kidnapping`, `robbery`, and `forced work stoppage`. Each template type is associated with the same set of 24 slots, which can be divided into **string-fill** slots — those that take (a set of) entities as fillers — and **set-fill** slots, which take a single filler from a fixed set of categorical values specific to each slot.[3] Table 1 shows dataset statistics and Appendix A lists all slots.

	Train	Dev	Test
Documents	1300	200	200
Sentences	18,317	2,989	2,702
Templates	1,114	191	209

Table 1: Statistics for the MUC-4 dataset. Sentence counts are based on our own sentence splitting methodology, as canonical sentence boundaries do not exist. Statistics are the same for languages in MULTIMUC.

Since the original MUC evaluations, it has become standard to evaluate systems on simplified templates that contain only string-fill slots (Chambers and Jurafsky, 2011; Du et al., 2021a,b; Chen et al., 2023b, *i.a.*), with the notable exception of the set-fill slot for template type. Additionally, while the gold data often lists multiple valid mentions for each entity filler, a system receives full credit for extracting just one of these. We follow both conventions in this work. The string-fill slots are `PerpInd` (individual perpetrators), `PerpOrg` (organizational perpetrators), `Target` (targeted infrastructure), `Weapon` (perpetrators' weapons), and `Victim` (victims of the event). Figure 1 shows a MUC-4 document and its simplified templates.

3 Data Collection

We now describe the data collection process for MULTIMUC, which consisted of four steps:

1. **Preprocessing** of the MUC-4 documents, including identification of sentence boundaries and locations of slot-filling entity mentions.
2. **Machine Translation** of the documents into each of the five target languages.
3. **Automatic Alignment** of slot-filling entity mentions in English with corresponding mentions in the target languages, followed by *projection* of the template annotations.
4. **Manual Correction** of entity mention alignments for all data splits, as well as translation

[1] https://openai.com/blog/chatgpt

[2] Following prior work (Du et al., 2021b; Chen et al., 2023b, *i.a.*), we will refer to template instances simply as *templates*.

[3] This is a minor simplification. See Appendix A.

Figure 2: Process for creating projected target language data for MULTIMUC from the gold (English) MUC-4 data.

corrections for sentences in the dev and test splits containing entity mentions.

Each step is detailed separately below. Figure 2 illustrates steps (1)-(3) for Farsi.

3.1 Preprocessing

We use the preprocessed version of the MUC-4 dataset released by Du et al. (2021b).[4] Three quirks of the dataset deserve mention.

First, to our knowledge, the documents were never released with canonical sentence splits. As such, we used an automatic tool, the Punkt sentence tokenizer from NLTK (Bird et al., 2009), to obtain sentence boundaries.[5]

Second, the text is uncased. This caused the sentence tokenizer to erroneously split a small number of sentences containing initialisms and titles (e.g. "u.s." or "dr.") into two or more fragments. We manually corrected these cases by searching on a fixed set of problematic terms (identified via manual inspection) and combining identified fragments.[6]

Third, character offsets of entity mentions are not annotated. This may be because evaluation has historically used string-based, rather than offset-based, matching to score string-fill slots. We follow Du et al. (2021b) in annotating the *first* occurrence of each mention string in a document and leave annotation of later occurrences for future work.

3.2 Machine Translation

Given the preprocessed English text, we obtain automatic translations of all 1,700 MUC-4 documents for all five of the target languages. Our MT system has a Stratified Mixture of Experts (SMoE) architecture (Xu et al., 2023) for multilingual translation. Mixture-of-experts (MoE) (Shazeer et al., 2017; Lepikhin et al., 2021) significantly scales up the number of parameters of multilingual neural MT transformer-based models while maintaining low computational requirements per token. SMoE enhances MoE models by assigning dynamic model capacity to different incoming tokens, hence enabling more efficient utilization of parameters. SMoE has demonstrated improvements over state-of-the-art MoE baselines (Xu et al., 2023).

We use an SMoE model pretrained on the primary bitexts of six languages from NLLB (Costa-jussà et al., 2022), covering over 70 million parallel sentences and all MULTIMUC languages.[7]

3.3 Automatic Alignment and Projection

Data projection involves automatically transferring span-level annotations from a source language to a target language based on word-to-word alignments. Given the translated documents, we first align each word in an English (source) sentence to the corresponding word(s) in the target sentence. Mentions in the target language are thus given by the sequence of target language tokens aligned to each token in an annotated source mention, and the corresponding slot and template in the source are thereby implicitly projected to the target.

We use Awesome-align (Dou and Neubig, 2021), an embedding-based word aligner that derives word alignments via comparison of word embeddings. Awesome-align fine-tunes a pretrained language model (in our case, XLM-R; Conneau et al., 2020) on parallel text or gold word alignments with objectives designed to improve alignment quality.

We reuse the models and empirically-chosen hyperparameters from prior work for a similar task

[4] https://github.com/xinyadu/gtt/

[5] https://www.nltk.org/_modules/nltk/tokenize/punkt.html. Punkt is based on the unsupervised, multilingual sentence tokenization algorithm of Kiss and Strunk (2006).

[6] The terms were *dr.*, *mr.*, *ms.*, *mrs.*, *gen.*, and *u.s.*

[7] The pretrained MT model can be downloaded from anonymous-url

(Zheng et al., 2023). These models are XLM-R encoders fine-tuned on around two million parallel target language-English sentences from the OSCAR corpus (Abadji et al., 2022). The encoders are further fine-tuned on gold alignments from GALE Chinese–English (Li et al., 2015), and the Farsi-English corpus by Tavakoli and Faili (2014), containing 2,800 Chinese–English and 1,200 Farsi-English sentence pairs with gold alignments. We further fine-tuned the model for Arabic on the 2,300 GALE Arabic-English (Li et al., 2013) sentence pairs with gold alignments.

3.4 Translation and Alignment Correction

While we find our automatic alignments to be of good quality (Table 2), prior work has shown that for some IE tasks, models can benefit meaningfully from access to gold alignments (Stengel-Eskin et al., 2019; Behzad et al., 2023). Accordingly, we recruited annotators to inspect and (if necessary) correct the automatic alignments for all sentences containing the first occurrence of some entity mention. Additionally, for the dev and test splits, annotators corrected the translations of these sentences.

Annotation was performed using a web app developed in-house for this purpose. Annotators were English speakers recruited from the authors' home institution, and all are also either native speakers of the language they annotated or are professional linguists with extensive training in that language. For practice, annotators completed 10 tasks that were not included in the final data. Given the annotators' level of competence as well as budgetary constraints, only a single annotator annotated each main task. Between one and four annotators worked on each language, with tasks distributed based on availability. Three of the annotators are authors of this work and were not paid; all others were paid at an average rate of $0.29 per task. Task instructions, examples of the interface, and some agreement statistics are given in Appendix B.

Entity and mention statistics for the training split of each language are shown in Table 2. In general, only a small fraction of the automatic alignments required correction: Even for the two languages requiring the most correction, Chinese and Russian, fully 77.4% of target language mentions were unchanged from the automatic alignment, rising to as much as 86.5% in the case of Arabic. This is testament to the quality of the alignments, though alignment quality is necessarily constrained by translation quality, discussed in Appendix B.

	Ar	Fa	Ko	Ru	Zh
Entities	2,421	2,432	2,417	2,394	2,071
$\text{Mentions}_{\text{man}}$	3,074	3,136	3,076	3,019	2,597
unchanged	86.5	84.0	79.7	77.4	77.4

Table 2: Entity and mention counts for the MULTIMUC training set. "$\text{Mentions}_{\text{man}}$" denotes *annotated* mentions. "Unchanged" denotes the percentage of $\text{Mentions}_{\text{man}}$ unchanged from the automatic alignment.

4 Experiments

We present three sets of experiments. All make use of the following three variations on training and dev data, designed to assess both the impact of alignment corrections and of parallel data:

1. **TGT_{AUTO}** uses only *target* language data, with mentions obtained via *automatic* alignments.
2. **TGT_{MAN}** uses only *target* language data, but with the *manually* corrected alignments for the training set and the corrected alignments and translations for the dev set.
3. **BI_{MAN}** is the same as TGT_{MAN}, but adds gold English (*bilingual*) training data.

In all experiments, we report results on the *annotated* test set.

4.1 Span Extraction

Setup Prior work investigating the impact of alignment quality in IE has focused on span labeling tasks such as NER or SRL (Stengel-Eskin et al., 2019; Behzad et al., 2023), as these tasks arguably give the most direct view on the downstream impact of improved alignments. In our first set of experiments, we follow this line of work and assess span extraction and labeling performance on MULTIMUC using the neural span extractor of (Xia et al., 2021), which has achieved state-of-the-art performance on FrameNet (Baker et al., 1998). We train the system to extract all slot-filling entity mentions and to label them with their slot.

Results Labeled and Unlabeled exact match F_1 scores for the three settings are shown in Table 3. Across almost all languages, we observe improvements on both metrics when training on corrected (TGT_{MAN}) vs. uncorrected (TGT_{AUTO}) data. Given that a fairly small proportion of spans in the data were changed between these settings, some of the gains may also be explained by access to corrected dev data in the TGT_{MAN} settings.

	Ar	Fa	Ko	Ru	Zh
TGT_{AUTO}	51.92	49.84	51.14	58.15	**54.46**
TGT_{MAN}	**56.25**	**55.62**	52.00	**59.34**	52.88
BI_{MAN}	54.89	53.34	**55.41**	57.40	53.44
TGT_{AUTO}	54.62	52.07	52.86	60.05	55.51
TGT_{MAN}	**58.88**	**56.82**	54.76	**62.54**	54.64
BI_{MAN}	56.60	55.10	**57.78**	59.66	**55.66**

Table 3: Labeled (top) and unlabeled (bottom) exact span match F_1 scores for all three data settings on the annotated test splits.

4.2 Template Filling with Fine-Tuned Models

Setup Our second set of experiments turns to template filling proper, focusing on the two models to have most recently achieved state-of-the-art on MUC-4. The first is GTT (Du et al., 2021b), which uses a single BERT-base model (Devlin et al., 2019) as both an encoder (to encode the document) and as a decoder, using causal masking and pointer decoding to generate linearized templates. As a minimal modification to support the MULTIMUC languages, we use *m*BERT-base (Devlin et al., 2019) in lieu of BERT-base, keeping all other aspects of the architecture unchanged.

The second model is ITERX (Chen et al., 2023b), which holds the current SOTA on MUC-4. ITERX treats template filling as autoregressive span classification, assigning each of a set of candidate spans (extracted by an upstream system) either to a slot in the current template or else to a special "null" slot to indicate that the span fills *no* slot in that template. Embeddings for the candidate spans are updated at each iteration based on their use in previous templates, and are used to condition the span assignments for subsequent templates. Chen et al. obtain their best MUC-4 results with a T5 encoder (Raffel et al., 2020). As with GTT, we make a minimal modification to the English base model by substituting *m*T5-base (Xue et al., 2021) for the encoder, keeping all else unchanged.[8]

Evaluation Evaluating template filling systems requires aligning predicted (P) and reference (R) templates, subject to the constraints that each reference template is aligned to at most one predicted one and that their types match. This is treated as a maximum bipartite matching problem, in which one seeks the alignment that yields a maximum total score over template pairs (P, R) given some template similarity function ϕ_T:

$$A^* = \operatorname*{argmax}_{A} \sum_{(P,R)\in A} \phi_T(P, R) \qquad (1)$$

$\phi_T(P, R)$ measures similarity between two templates in terms of similarity of their slot fillers, and there are different ways this can be done. Du et al. (2021b) propose the CEAF-REE metric, which computes an optimal alignment between predicted and reference *entities* similar to the CEAF metric for coreference resolution (Luo, 2005), but within slot. CEAF-REE selects the template alignment that yields the highest micro-F_1 over all slot fills, *including template type*. However, Chen et al. (2023b) take issue with certain properties of CEAF-REE and propose a variant called CEAF-R*ME*. The key differences from CEAF-REE are (1) template type is *excluded* from the F_1 calculation and (2) a different similarity function is used for computing entity alignments. We report both metrics and refer the reader to their paper for further details.[9]

Results Results for all languages are presented in the first six rows of Table 4. Several observations stand out. First, for nearly all languages, both models obtain their strongest performance when trained jointly on English and target language data (BI_{MAN}). This is consistent with past findings in IE establishing the value of English training data for less resourced target languages (Subburathinam et al., 2019; Yarmohammadi et al., 2021; Fincke et al., 2022, *i.a.*). While the impact of the English data is valuable for both models, it is especially so for ITERX, for which it boosts performance relative to the next best setting by an average of about 8.3 CEAF-REE F_1 and an average of over 4.7 CEAF-RME F_1 (compared to 3.2 and 2.6 F_1 for GTT).

Second, the benefits of training on the target language data with corrected alignments (TGT_{MAN}) are most evident for GTT, for which it shows uniform improvements relative to no corrections (TGT_{AUTO}) for CEAF-RME scores.[10] In contrast, performance does not substantially differ between the two settings for ITERX. This may be a consequence of

[8] We stress that our interest here is to present the best results for each model type and to evaluate cross-lingual performance variation *within* type, not in cross-type comparisons. For a comparison on MUC-4 of ITERX and GTT under identical encoders, see Chen et al. (2023b). Additional details on architectures and hyperparameters are provided in Appendix C.

[9] In Chen et al.'s terminology, we report $\text{CEAF-REE}_{\text{impl}}$ and CEAF-RME_{ϕ_3}.

[10] CEAF-*REE* scores are expected to show a noisier relationship with alignment correction due to the inclusion of the template type slot in the F_1 calculation, as accuracy is usually much higher for this slot than for others.

ITERX's reliance on an upstream system for its candidate spans: to isolate the effect of ITERX *training*, these candidates were fixed across settings at inference time, but it's quite plausible that the added value of corrected alignments lies chiefly in the span extraction step (see §4.1).

Lastly, the best scores for both models in all five MULTIMUC langauges are low by comparison to the best reported results on English. There is clear room for improvement across all languages, and we are excited by the prospect of better models more tailored to specific languages.

4.3 Few-Shot Template Filling

With the staggering leaps in the capabilities of large (and especially proprietary) language models of the past couple years, an immediate question for most tasks asks how competitive these models are in a zero- or few-shot setting compared to smaller, fine-tuned models (§4.2). We consider this question for MULTIMUC, investigating the capabilities of ChatGPT[11] on few-shot template filling. While ChatGPT's training corpus is predominantly English, already some works have studied its abilities on MT (Jiao et al., 2023; Peng et al., 2023) and on IE tasks in other languages (Lai et al., 2023), and found solid results. To our knowledge, this is the first work exploring few-shot template filling *at all*.

Setup We use the long-context version of ChatGPT (`gpt-3.5-turbo-16k-0613`) and evaluate in the TGT_{MAN} and BI_{MAN} settings. The system prompt informs the model that it is an expert in IE and that it must perform extraction on a target document. The user prompt provides more detailed instructions, including the desired output format for extracted templates, as well as three examples of other documents with their gold templates.[12] For the TGT_{MAN} setting, example documents are chosen from the target language training set using a BM25 retrieval model and are sorted so that the most relevant example is last. For the BI_{MAN} setting, we replace the most relevant target language example with the corresponding English one.

Results Results are shown in the bottom two rows of Table 4. Performance in both settings trails the performance of ITERX and GTT across languages — a finding in line with prior work showing that ChatGPT's few-shot capabilities on many tasks still fall short of those of the best supervised models (Lai et al., 2023; Gao et al., 2023), and an unsurprising result given its predominantly English training corpus. Furthermore, the clear gains from English training data for the supervised models do not clearly carry over here: including a relevant English document in the prompt helps only in some cases and even then only modestly.

[11] https://openai.com/blog/chatgpt

[12] Some effort was invested in identifying effective prompts for this task, but our aim here is *not* an extensive prompt engineering project, but rather a reasonable baseline. Prompt examples and hyperparameter details are in Appendix C.

5 Discussion

Here we present some analysis of model errors (§5.1) and also discuss observations and challenges from annotation (§5.2).

5.1 Model Errors

We use the template filling error analysis tool of Das et al. (2022) to understand the distribution of error types in the predictions from GTT.[13] Das et al. define a set of transformations by which a set of predicted templates may be converted into the gold ones, given an optimized template alignment (see §4). These include insertion and deletion transformations for templates and role fillers, as well as edit transformations for mentions and their role assignments. Error types are then defined in terms of transformation *sequences*.

Figure 3 shows a breakdown of errors by type for all languages and all three data settings for GTT. Consistent with Das et al.'s observations for MUC-4, we find that across languages and settings, missing role fillers account for a majority of the errors.[14] This is unsurprising when considering both that GTT's extractions heavily favor precision (Du et al., 2021b) and that models generally tend to struggle significantly with template recall, perhaps due to difficulty in *individuating* events (Gantt et al., 2022). Spurious templates and role fillers represent a smaller but non-trivial fraction of all errors.

5.2 Annotation Observations

We now discuss observations and challenges from the annotation process. While there are obviously many language-specific considerations for both translation and alignment, we highlight several that were common to two or more languages.

[13] Source code for the tool can be found here: https://github.com/IceJinx33/auto-err-template-fill/

[14] This includes both "Missing Role Filler" errors (i.e. role fillers missing from a predicted template) and "Missing Template Role Filler" errors (i.e. role fillers missing due to the associated template not being predicted in the first place).

		CEAF-REE						CEAF-RME					
		En	Ar	Fa	Ko	Ru	Zh	En	Ar	Fa	Ko	Ru	Zh
GTT	TGT$_{\text{AUTO}}$		24.26	31.46	34.17	35.38	36.74		11.27	16.24	18.24	20.23	18.90
	TGT$_{\text{MAN}}$	50.23	28.81	36.01	33.79	**38.05**	36.35	32.30	15.05	21.27	18.71	**22.44**	19.11
	BI$_{\text{MAN}}$		**36.76**	**37.91**	**36.52**	36.97	**41.48**		**21.98**	**22.44**	**20.71**	21.26	**23.26**
ITERX	TGT$_{\text{AUTO}}$		25.55	27.15	25.99	29.61	27.54		15.96	17.78	16.52	19.58	17.60
	TGT$_{\text{MAN}}$	53.00	25.70	25.36	27.24	30.08	27.32	35.20	15.73	16.41	17.11	19.30	17.06
	BI$_{\text{MAN}}$		**34.73**	**33.15**	**37.02**	**36.95**	**36.02**		**21.46**	**20.66**	**23.91**	**23.77**	**21.93**
CHATGPT	TGT$_{\text{MAN}}$	29.11	23.77	21.02	**17.14**	**25.40**	23.36	22.41	14.67	12.91	6.73	**16.38**	**15.02**
	BI$_{\text{MAN}}$		**24.62**	**22.06**	16.85	24.90	**24.46**		**14.79**	**13.42**	**7.12**	15.36	13.99

Table 4: CEAF-REE and CEAF-RME F_1 scores on English and the five MULTIMUC languages for GTT (Du et al., 2021b), ITERX (Chen et al., 2023b), and CHATGPT under the data settings described in §4. English results are the best ones reported in (Chen et al., 2023b), except for CHATGPT, and do not correspond to any of the three data settings. **Bolded** results are best results within model type. See §4.2 for caveats about cross-type comparisons.

5.2.1 Proper Nouns

MUC-4 annotations contain a significant number of proper nouns with a single canonical form, and these were sometimes translated into multiple forms in the target language, including both acceptable variants (e.g. the Farsi "هتل شراتن" [hoh-tel she-raa-tohn] or "هتل شرایتن" [hoh-tel she-reye-tohn] for "Sheraton Hotel") and orthographic errors (레이 [ɭe.i], 릴리 [ɭiɭ.ɭi], or 릴 [ɭiɭ] for the name "Leigh"). In Chinese, each syllable in a proper noun may be translated into one of several characters that approximate the pronunciation. E.g., the first syllable of "Guatemala" may phonetically correspond to 危 [wēi] or 瓜 [guā], and the noun as a whole can be translated as either 危地拉 or 瓜地拉. These forms were canonicalized as much as possible in the dev and test annotations, but this could not be done for train by virtue of the annotation protocol.

5.2.2 Word Order

In general, Farsi has subject-object-verb word order and Arabic has verb-subject-object order. However, in both languages, the order can sometimes change because of the context, certain case endings, and adverbs. In a number of instances, annotators noted that the automatic translations use the standard word order even when changing it would result in a more natural phrasing and corrected these cases. As an example, for the sentence "the rebels who (...) attacked the building", the automatic Arabic translation was "هاجم المتمردون الذين (...) المبنى", where "هاجم" is the verb, "المتمردون" is the subject and "المبنى" is the object. But a more natural-sounding translation would be "المتمردون الذين (...) هاجموا المبنى".

5.2.3 Numeral classifiers

Chinese and Korean mark nouns with classifiers (CL) when naming and counting them. In both languages, a CL always follows a numeral when an explicit number is present, and in Korean, when the combination of a numeral and a CL follows its associated noun, aligning the classifier to the noun is less desirable, as this would result in discontiguous target language spans. As such, annotators aligned numerals in English to both the numeral and CL in the target languages, as illustrated in Example (1). Relatedly, for Chinese translation correction, annotators combined a (numeral, CL) pair into one token when they were translated as separate tokens.

(1)	경찰	세	명	(Korean)
	gyeongchal	***se***	**myeong**	
	policeman	**three**	**CL**	
	'**three** policemen'			

6 Related Work

Template Filling Template filling has a long history. Participants in the MUCs, starting with MUC-3 (muc, 1991) and MUC-4 (muc, 1992), largely developed pipelined, rule-based systems with individual modules designed to solve problems that are now major NLP tasks in their own right, such as coreference resolution and semantic role labeling (Hobbs, 1993; Grishman, 2019). MUC-5 introduced a considerably more complicated template ontology that represented entities *themselves* as templates, yielding nested template structures (muc, 1993). MUC-6 (muc, 1995) and MUC-7 (muc, 1998) also featured nested templates, though the entity templates were pared down to fewer slots and there was only a single event type of interest.

Following the MUCs, many works revisiting these corpora focused on *role-filler entity extraction*, a simplified form of template filling in which the goal is to identify all entity fillers, but without

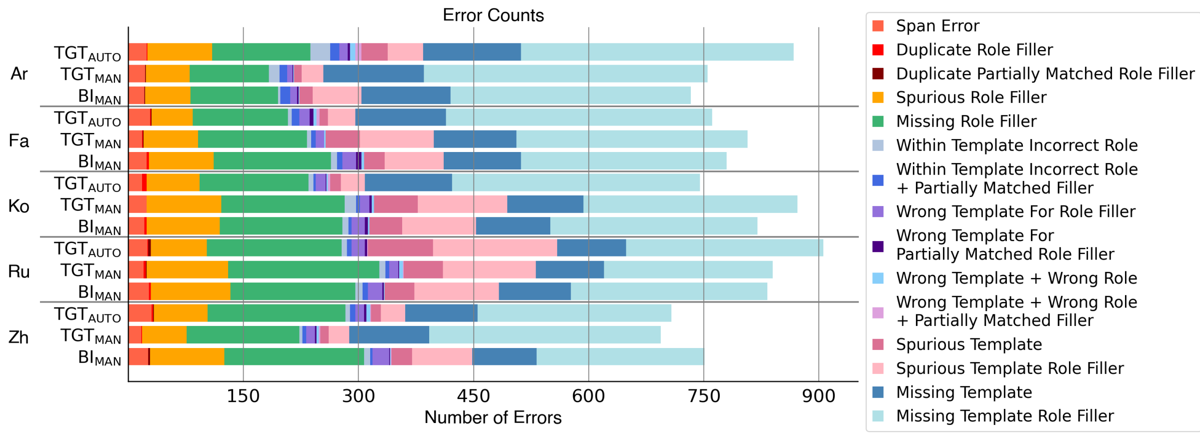

Figure 3: Automated error analysis results based on the error analysis tool of Das et al. (2022) for GTT test set predictions for all MultiMUC languages and all data settings (see §4). Missing role filler errors predominate.

collating them into distinct templates (Patwardhan and Riloff, 2007, 2009; Huang and Riloff, 2011, 2012; Du et al., 2021a; Huang et al., 2021).

Note that template filling differs from document-level N-ary relation extraction in being *event-centric* and in allowing null arguments. It differs from event extraction in not having event triggers.

Multilingual Template Filling Works cited in preceding sections (Du et al., 2021b; Chen et al., 2023b; Das et al., 2022) exhaust deep learning-era efforts on template filling with MUC-4. Even as early as the MUC-4 conference itself, though, there was interest in extending template filling systems to other languages. NYU's PROTEUS system, for instance, was extended to handle Spanish documents (Grishman et al., 1992), and the SOLOMON system from Systems Research and Applications (SRA) was enhanced to handle both Spanish and Japanese documents (Aone et al., 1992, 1993). This work presaged MUC-5, which had evaluations in both English and Japanese, but as best we know, no corpora were ever released for either language.

More recently, Zheng et al. (2019) used distant supervision techniques to construct the ChFinAnn template filling dataset, which contains roughly 32,000 Chinese news articles annotated for five finance-related event types, though this dataset is monolingual. More similar to MultiMUC, the IARPA BETTER program (Soboroff, 2023) introduced the BETTER Granular dataset with an ontology of six diverse template types (e.g. protests, epidemics, natural disasters), covering news articles in English and five other languages. Granular is notable as the only multilingual template filling dataset that has both gold document texts and gold template annotations, though this is not parallel data and the corpus is much smaller than MUC-4, with only several hundred documents.

Cross-Lingual Alignment and Projection Cross-lingual projection is a method for transferring annotations from a source language to a target language, used primarily to create cross-lingual datasets for structured prediction tasks (Yarowsky and Ngai, 2001; Aminian et al., 2019; Fei et al., 2020; Daza and Frank, 2020; Ozaki et al., 2021; Yarmohammadi et al., 2021; Chen et al., 2023a, *i.a.*). The approach relies on two main steps: translation and source-to-target word alignment, and thus relies on high-quality translations and alignments between source and target texts. Studies have shown that access to gold entity alignments can improve downstream results (Stengel-Eskin et al., 2019; Behzad et al., 2023).

7 Conclusion

We have introduced MultiMUC— to our knowledge the first multilingual *parallel* template filling dataset, featuring high-quality automatic translations of the MUC-4 corpus along with human translations of key portions of the dev and test splits, and human-annotated alignments for all fillers of string-fill slots. Moreover, we have established strong mono- and bilingual baselines using two recent, top-performing template filling models, as well as baselines for few-shot template filling — seemingly the first few-shot evaluations for this task. Lastly, we have highlighted some observations and challenges involved in constructing this resource and presented a detailed breakdown of model errors. We hope that this work will facilitate further research on multilingual IE at the document level.

Limitations

Ideally, all datasets that include machine-generated outputs would have exhaustive human verification and correction of those outputs. This of course applies to MULTIMUC: while the dataset provides human translations of key portions of the dev and test splits (those containing the first occurrence of each entity mention), the majority of sentences in the dataset are machine-translated, which does result in a small number of data projection failures (see Appendix B). We intend to obtain gold translations and entity alignments for the entire corpus in follow-up work, but this was infeasible with the personnel and budget available to us for the present work. Regardless, the automatic alignments and translations are of good quality (see §3 and Appendix B) and make MULTIMUC a valuable resource for training and evaluating document-level IE systems in multiple languages.

Ethics Statement

We do not believe this work raises significant ethical concerns.

References

1991. *Third Message Understanding Conference (MUC-3): Proceedings of a Conference Held in San Diego, California, May 21-23, 1991*.

1992. Appendix A: Evaluation task description. In *Fourth Message Understanding Conference (MUC-4): Proceedings of a Conference Held in McLean, Virginia, June 16-18, 1992*.

1992. *Fourth Message Understanding Conference (MUC-4): Proceedings of a Conference Held in McLean, Virginia, June 16-18, 1992*.

1993. *Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland, August 25-27, 1993*.

1995. *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*.

1998. *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*.

Julien Abadji, Pedro Ortiz Suarez, Laurent Romary, and Benoît Sagot. 2022. Towards a cleaner document-oriented multilingual crawled corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4344–4355, Marseille, France. European Language Resources Association.

Maryam Aminian, Mohammad Sadegh Rasooli, and Mona Diab. 2019. Cross-lingual transfer of semantic roles: From raw text to semantic roles. In *Proceedings of the 13th International Conference on Computational Semantics - Long Papers*, pages 200–210, Gothenburg, Sweden. Association for Computational Linguistics.

Chinatsu Aone, Hatte Blejer, Sharon Flank, Douglas McKee, and Sandy Shinn. 1993. The Murasaki project: Multilingual natural language understanding. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 21-24, 1993*.

Chinatsu Aone, Doug McKee, Sandy Shinn, and Hatte Blejer. 1992. SRA Solomon: MUC-4 test results and analysis. In *Fourth Message Understanding Conference (MUC-4): Proceedings of a Conference Held in McLean, Virginia, June 16-18, 1992*.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Shabnam Behzad, Seth Ebner, Marc Marone, Benjamin Van Durme, and Mahsa Yarmohammadi. 2023. The effect of alignment correction on cross-lingual annotation projection. In *Proceedings of the 17th Linguistic Annotation Workshop (LAW-XVII)*, pages 244–251, Toronto, Canada. Association for Computational Linguistics.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Nathanael Chambers and Dan Jurafsky. 2011. Template-based information extraction without the templates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 976–986, Portland, Oregon, USA. Association for Computational Linguistics.

Yang Chen, Chao Jiang, Alan Ritter, and Wei Xu. 2023a. Frustratingly easy label projection for cross-lingual transfer. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 5775–5796, Toronto, Canada. Association for Computational Linguistics.

Yunmo Chen, William Gantt, Weiwei Gu, Tongfei Chen, Aaron White, and Benjamin Van Durme. 2023b. Iterative document-level information extraction via imitation learning. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1858–1874, Dubrovnik, Croatia. Association for Computational Linguistics.

Jackie Chi Kit Cheung, Hoifung Poon, and Lucy Vanderwende. 2013. Probabilistic frame induction. In *Proceedings of the 2013 Conference of the North*

American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 837–846, Atlanta, Georgia. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Aliva Das, Xinya Du, Barry Wang, Kejian Shi, Jiayuan Gu, Thomas Porter, and Claire Cardie. 2022. Automatic error analysis for document-level information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3960–3975, Dublin, Ireland. Association for Computational Linguistics.

Angel Daza and Anette Frank. 2020. X-SRL: A parallel cross-lingual semantic role labeling dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3904–3914, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2112–2128, Online. Association for Computational Linguistics.

Xinya Du, Alexander Rush, and Claire Cardie. 2021a. GRIT: Generative role-filler transformers for document-level event entity extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 634–644, Online. Association for Computational Linguistics.

Xinya Du, Alexander Rush, and Claire Cardie. 2021b. Template filling with generative transformers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 909–914, Online. Association for Computational Linguistics.

Hao Fei, Meishan Zhang, and Donghong Ji. 2020. Cross-lingual semantic role labeling with high-quality translated training corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7014–7026, Online. Association for Computational Linguistics.

Steven Fincke, Shantanu Agarwal, Scott Miller, and Elizabeth Boschee. 2022. Language model priming for cross-lingual event extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10627–10635.

William Gantt, Reno Kriz, Yunmo Chen, Siddharth Vashishtha, and Aaron Steven White. 2022. On event individuation for document-level information extraction. *arXiv preprint arXiv:2212.09702*.

Jun Gao, Huan Zhao, Changlong Yu, and Ruifeng Xu. 2023. Exploring the feasibility of chatgpt for event extraction. *arXiv preprint arXiv:2303.03836*.

Ralph Grishman. 2019. Twenty-five years of information extraction. *Natural Language Engineering*, 25(6):677–692.

Ralph Grishman, John Sterling, and Catherine Macleod. 1992. New York University PROTEUS system: MUC-4 test results and analysis. In *Fourth Message Understanding Conference (MUC-4): Proceedings of a Conference Held in McLean, Virginia, June 16-18, 1992*.

Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference- 6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Jerry R. Hobbs. 1993. The generic information extraction system. In *Fifth Message Understanding Conference (MUC-5): Proceedings of a Conference Held in Baltimore, Maryland, August 25-27, 1993*.

Kung-Hsiang Huang, Sam Tang, and Nanyun Peng. 2021. Document-level entity-based extraction as template generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5257–5269, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ruihong Huang and Ellen Riloff. 2011. Peeling back the layers: Detecting event role fillers in secondary contexts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1137–1147, Portland, Oregon, USA. Association for Computational Linguistics.

Ruihong Huang and Ellen Riloff. 2012. Modeling textual cohesion for event extraction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 26, pages 1664–1670.

Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Xing Wang, and Zhaopeng Tu. 2023. Is chatgpt a good translator? a preliminary study. *arXiv preprint arXiv:2301.08745*.

Tibor Kiss and Jan Strunk. 2006. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525.

Viet Dac Lai, Nghia Trung Ngo, Amir Pouran Ben Veyseh, Hieu Man, Franck Dernoncourt, Trung Bui, and Thien Huu Nguyen. 2023. Chatgpt beyond english: Towards a comprehensive evaluation of large language models in multilingual learning. *arXiv preprint arXiv:2304.05613*.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021. Gshard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations (ICLR)*.

Xuansong Li, Stephen Grimes, Safa Ismael, Stephanie Strassel, Mohamed Maamouri, and Ann Bies. 2013. GALE arabic-english parallel aligned treebank – broadcast news part 1. LDC2013T14.

Xuansong Li, Stephen Grimes, Stephanie Strassel, Xiaoyi Ma, Nianwen Xue, Mitchell P. Marcus, and Ann Taylor. 2015. GALE chinese-english parallel aligned treebank – training. LDC2015T06.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Hiroaki Ozaki, Gaku Morio, Terufumi Morishita, and Toshinori Miyoshi. 2021. Project-then-transfer: Effective two-stage cross-lingual transfer for semantic dependency parsing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2586–2594, Online. Association for Computational Linguistics.

Siddharth Patwardhan and Ellen Riloff. 2007. Effective information extraction with semantic affinity patterns and relevant regions. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 717–727, Prague, Czech Republic. Association for Computational Linguistics.

Siddharth Patwardhan and Ellen Riloff. 2009. A unified model of phrasal and sentential evidence for information extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 151–160, Singapore. Association for Computational Linguistics.

Keqin Peng, Liang Ding, Qihuang Zhong, Li Shen, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. 2023. Towards making the most of chatgpt for machine translation. *arXiv preprint arXiv:2303.13780*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Ian Soboroff. 2023. The better cross-language datasets. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3047–3053.

Elias Stengel-Eskin, Tzu-ray Su, Matt Post, and Benjamin Van Durme. 2019. A discriminative neural model for cross-lingual word alignment. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 910–920, Hong Kong, China. Association for Computational Linguistics.

Ananya Subburathinam, Di Lu, Heng Ji, Jonathan May, Shih-Fu Chang, Avirup Sil, and Clare Voss. 2019. Cross-lingual structure transfer for relation and event extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 313–325, Hong Kong, China. Association for Computational Linguistics.

Leila Tavakoli and Heshaam Faili. 2014. Phrase alignments in parallel corpus using bootstrapping approach. *International Journal of Information and Communication Technology*, 6:63–76.

Patrick Xia, Guanghui Qin, Siddharth Vashishtha, Yunmo Chen, Tongfei Chen, Chandler May, Craig Harman, Kyle Rawlins, Aaron Steven White, and Benjamin Van Durme. 2021. LOME: Large ontology multilingual extraction. In *Proceedings of the 16th*

Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, pages 149–159, Online. Association for Computational Linguistics.

Haoran Xu, Maha Elbayad, Kenton Murray, Jean Maillard, and Vedanuj Goswami. 2023. Towards being parameter-efficient: A stratified sparsely activated transformer with dynamic capacity.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Mahsa Yarmohammadi, Shijie Wu, Marc Marone, Haoran Xu, Seth Ebner, Guanghui Qin, Yunmo Chen, Jialiang Guo, Craig Harman, Kenton Murray, Aaron Steven White, Mark Dredze, and Benjamin Van Durme. 2021. Everything is all it takes: A multipronged strategy for zero-shot cross-lingual information extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1950–1967, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

David Yarowsky and Grace Ngai. 2001. Inducing multilingual POS taggers and NP bracketers via robust projection across aligned corpora. In *Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Boyuan Zheng, Patrick Xia, Mahsa Yarmohammadi, and Benjamin Van Durme. 2023. Multilingual Coreference Resolution in Multiparty Dialogue. *Transactions of the Association for Computational Linguistics*, 11:922–940.

Shun Zheng, Wei Cao, Wei Xu, and Jiang Bian. 2019. Doc2EDAG: An end-to-end document-level framework for Chinese financial event extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 337–346, Hong Kong, China. Association for Computational Linguistics.

A MUC-4 Template Slots

Below is the complete list of MUC-4 slots, which are the same for all template types, along with their definitions as provided in the conference appendices (nn-, 1992).[15] The names of the string-fill slots are **bolded** and their (more commonly used) alternative names are given in parentheses. The significant majority of others are set-fill, though some slots require a numerical answer (e.g. "PHYS TGT: NUMBER") and these are known as *text conversion* slots, as they require *converting* possibly implicit counts of entities in the text into explicit numerical values. We group these with set-fill slots in the main text as they have likewise traditionally been excluded from evaluation since the original conference. "MESSAGE: ID" and "MESSAGE: TEMPLATE" were never part of the evaluation, even in the original conference. Some of the slot names use one or more of the following abbreviations: PERP = perpetrator; PHYS = physical; TGT = target; HUM = human.

1. MESSAGE: ID — The first line of the message, e.g., DEV-MUC3-0001 (NOSC). This slot serves as an index and is not scored in its own right.
2. MESSAGE: TEMPLATE — A number that distinguishes the templates for a given message. In the answer key, the word OPTIONAL in parentheses after the template number indicates that there is significant doubt whether the incident belongs in the database.
3. INCIDENT: DATE — The date of incident (according to local time, not Greenwich Mean Time).
4. INCIDENT: LOCATION — The place where the incident occurred.
5. INCIDENT: TYPE — A terrorist act reported on in the message.
6. INCIDENT: STAGE OF EXECUTION — An indicator of whether the terrorist act was accomplished, attempted, or merely threatened.
7. **INCIDENT: INSTRUMENT ID** (`Weapon`) — A device used by the perpetrator(s) in carrying out the terrorist act.
8. INCIDENT: INSTRUMENT TYPE — The category that the instrument fits into.
9. PERP: INCIDENT CATEGORY — The subcategory of terrorism that the incident fits into, as determined by the nature of the perpetrators.
10. **PERP: INDIVIDUAL ID** (`PerpInd`) — A person responsible for the incident.
11. **PERP: ORGANIZATION ID** (`PerpOrg`) — An organization responsible for the incident.
12. PERP: ORGANIZATION CONFIDENCE — The way a perpetrator organization is viewed in the message.
13. **PHYS TGT: ID** (`Target`) — A thing (inanimate object) that was attacked.
14. PHYS TGT: TYPE — The category that the physical target fits into.
15. PHYS TGT: NUMBER — The number of physical targets with a particular ID and TYPE.
16. PHYS TGT: FOREIGN NATION — The nationality of a physical target, if the nationality is identified in the article and if it's different from country where incident occurred.
17. PHYS TGT: EFFECT OF INCIDENT — The impact of the incident on a physical target.
18. PHYS TGT: TOTAL NUMBER — The total number of physical targets.

[15]The original MUC-3 and MUC-4 data can be found at the following URL: `https://www-nlpir.nist.gov/related_projects/muc/muc_data/muc_data_index.html`. The licit set of values for each set-fill slot can also be found in (nn-, 1992). While the slots are the same across template types, the licit values of some set-fill slots are type-dependent.

19. **HUM TGT: NAME** `(Victim)` — The name of a person who was the obvious or apparent target of the attack or who became a victim of the attack.

20. **HUM TGT: DESCRIPTION** — The title or role of a named human target or a general description of an unnamed human target.

21. HUM TGT: TYPE — The category that the human target fits into.

22. HUM TGT: NUMBER – The number of human targets with a particular NAME, DESCRIPTION, and TYPE.

23. HUM TGT: FOREIGN NATION – The nationality of a human target, if the nationality is identified in the article and if it's different from country where incident occurred.

24. HUM TGT: EFFECT OF INCIDENT – The impact of the incident on a human target(s).

25. HUM TGT: TOTAL NUMBER – The total number of human targets.

B Data Collection

This appendix presents additional details about our data collection procedure, including the instructions that were provided to annotators (§B.1), screenshots of the annotation interface (§B.2), and some measures and discussion of data quality (§B.3).

All annotators were told about the broad goals of the project prior to starting the task and were told that their annotations would be used for this project. The trained linguists who provided annotations are employees or contractors of the authors' home institution who are paid a regular salary for annotation work, though we (the authors) were not informed of the exact salary of each annotator. Some of the native speaker annotators were authors of the paper and were not paid, as mentioned in §3; others were undergraduate students at the same institution, recruited through an internal job posting. The $0.29 per-task pay rate given in the main text was computed by dividing the total pay for student annotators for each language ($720) by the total number of tasks for each language (2,450). All annotation has been approved by the authors' home institution.

B.1 Task Instructions

Below are the task instructions that were presented to all annotators.

Overview

In each task, a pair of sentences, one in English ("source") and one in another ("target") language will be shown to the user. The English sentence will be shown on the top half of the screen and an automatic translation of the English sentence into the target language will be shown on the bottom half. Both sentences will be segmented into words ("tokenized"). The task is to verify and correct alignments between highlighted spans of English text (each consisting of one or more words) and their translations in the target language. In each English sentence, there will typically be more than one span to align. The user needs to annotate the English spans word by word. By clicking on each English word, a *suggested* span in the target language, based on an automatic ("default") alignment between words in the English and target language sentences, is highlighted as the default answer on the target side (bottom of the screen). In some cases, you may also have the option to correct the target language translation as well.

Instructions

The default alignment

- If you think the default alignment is correct (and the translation, if correcting the translation), simply press "submit."

- If you want to modify the default alignment, select the corresponding source span, modify the target span, and press "submit."

Aligning spans

- Only the source spans we are interested in are highlighted. All other words in the source sentence are greyed out.

- While ideally aligned spans in the target language will consist of contiguous sequences of words, it's OK to select non-contiguous target words if appropriate.

- It may sometimes be the case either that (1) a word in the English does not have any clear analogue in the target language, or (2) a word in the target language does not have any clear analogue in English. In these cases, you can do one of two things.
  - One possibility is to align the word without a clear analogue to a closely related word. For instance, "happiness" in English is translated in French as "le bonheur," where "le" is a definite article, which is not used in the English. Here, we would align "le" to "happiness," since it's part of a multi-word expression that denotes the same thing as "happiness" does. In general, this solution should be preferred.
  - Another possibility is to simply remove the word from the alignment. In general, this should be done only if the word is *not* part of a multi-word expression (unlike "le" in "le bonheur" above) or seems like a translation error (that you cannot correct; see **Retokenizing the target sentence**).

- As we are not experts in most of the languages we are annotating here, you will likely encounter other difficult alignment decisions we have not foreseen. When you first encounter such instances, try to formulate general rules that seem sensible to you and apply them consistently throughout the rest of your annotation.

Retokenizing the target sentence

- If you see the "RE-TOKENIZE" button on the target side, you are allowed to edit the target side text to correct the potential mistakes in automatic translation or word segmentation. When correcting translations, you should correct ALL text in the sentence that needs it — not just the tokens highlighted by the default alignments. You are allowed to edit or remove existing tokens, add new words, or split or merge the existing words to correct word segmentation. When retokenizing, each word or punctuation mark should go on its own line.

- If you make changes using "RE-TOKENIZE," the suggested target spans will be automatically adjusted. In general, this adjustment should be correct: any words on the target side that you did not change should remain aligned to the correct word on the source side, even if you insert or delete other words. Of course, if you delete an aligned word on the target side, alignments to that word will be removed. Importantly, the same will be the case if you edit an aligned word, so you will have to realign any edited words. If you do make changes using "RE-TOKENIZE," you should always double-check that the alignments are correct before submitting.

Mistakes

- Finally, if you make a mistake during annotation or encounter a technical problem in the interface, please try to note down the ID of the task you are working on at the time and inform us of the mistake or problem. The Task ID can be found in the top right corner of the screen ("Task ID: ⟨#⟩"). **Please get in the habit of noting the task ID as soon as you accept it!**
  - **NOTE**: We have noticed that some workers accidentally click the submit button after re-tokenizing, when they mean to click the **save** button (to save their new tokenization). Please try to avoid doing this, but tell us if you do.

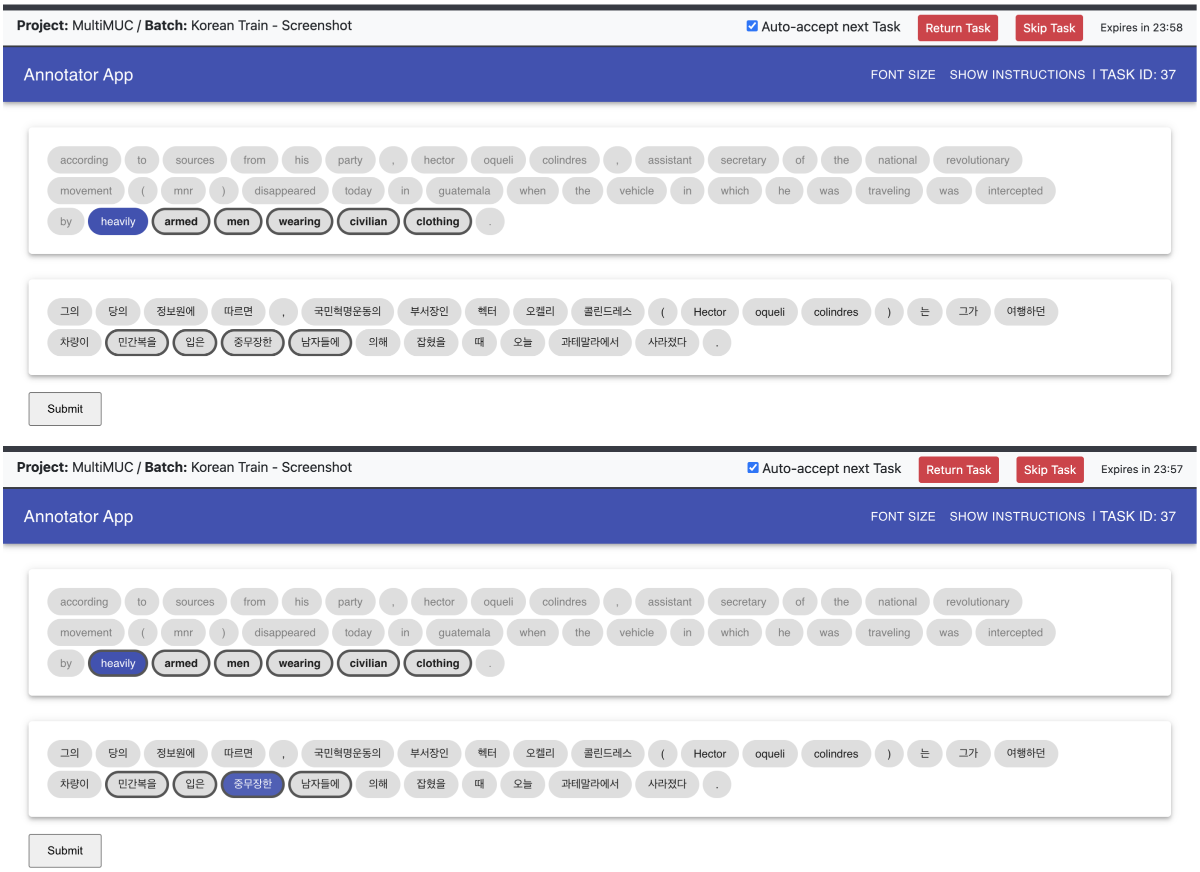

Figure 4: A Korean training split task before (top) and after (bottom) manual alignment correction.

B.2 Task Interface

Recall from §3 that alignment corrections were collected for all three splits (train, dev, and test) and that translation corrections were collected for the dev and test splits only. The same interface was used for both types of annotation. Figure 4 and Figure 5 show examples of the interface for Korean annotation. Figure 4 shows the interface as it appears when doing alignment correction only (i.e. training set annotation), both before any alignment correction (top) and after (bottom). Figure 5 shows the interface as it appears when *also* doing translation correction (i.e. dev and test set annotation) — once again both before correction (top) and after (bottom). The only difference in the interface between the two figures is the presence of the "RE-TOKENIZE" button in Figure 5, which, when clicked, allows annotators to change (insert/edit/delete) target language tokens. In both cases, when a new task is loaded, the annotator sees a "default alignment," which is simply the automatic token alignment that is obtained using Awesome-align (Dou and Neubig, 2021) and that is in the TGT_{AUTO} experiments. This is the alignment they must correct.

B.3 Data and Annotation Quality

As discussed in §3, our annotators were all either native speakers of the language they annotated or else were linguists with significant formal training in that language. Given this, and given that effective alignment and translation correction require only linguistic competence, the quality of the annotations can be presumed to be very high.

Even so, we provide some limited quantitative measures of annotation quality. We first report inter-annotator agreement on alignment correction for Farsi and Chinese for a randomly selected 50 tasks from the training set. We report Cohen's κ at the token level: two alignments for a particular English token count as equivalent iff they align exactly the same target language token(s) to that English token. Two annotators completed these tasks for each language. For Farsi, we obtained a κ of 0.98. For Chinese, we

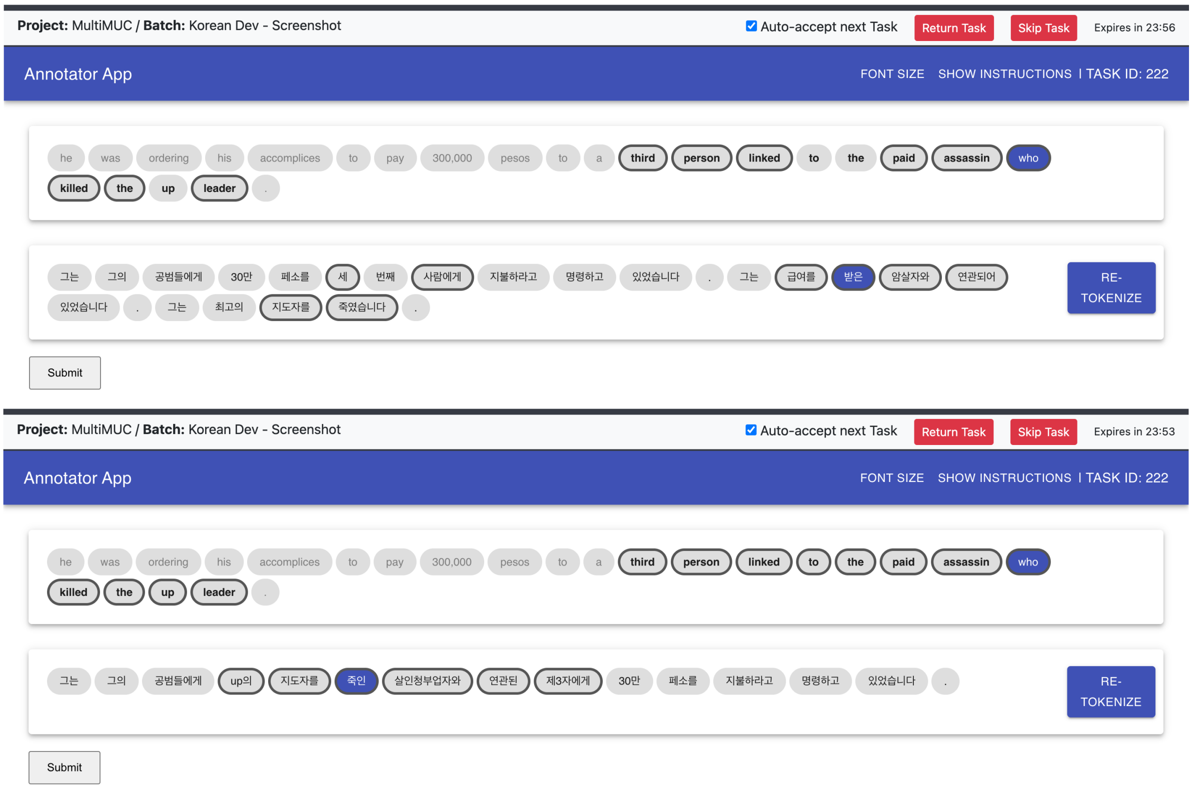

Figure 5: A Korean dev split task before (top) and after (bottom) manual alignment *and* translation correction.

obtained a κ of 0.87. Both indicate "almost perfect" agreement.[16]

We additionally report sacreBLEU scores (Post, 2018) between the uncorrected and corrected dev and test data for all languages to give a more quantitative sense of how similar the translation corrections are to the original, machine-translated text. The BLEU scores on the combined dev and test sets for Arabic, Farsi, Korean, Russian, and Chinese are (respectively) 73.1, 83.6, 76.1, 89.3, and 65.2. BLEU scores higher than 60 are often considered "better than human"[17] and imply that the uncorrected and corrected translations can be considered as translations of the same source.

Finally, as we allude to in the limitations section, due to the lack of translation correction for the training set, translation errors resulted in a small fraction of entity mentions (and sometimes entities) failed to be aligned and projected from the English. This included 4.6% of mentions (and 3.2% of entities) for Arabic, 3.0% of mentions (2.4% of entities) for Farsi, 4.4% of mentions (3.1% of entities) for Korean, 6.9% of mentions (4.1% of entities) for Russian, and 17.7% of mentions (15.6% of entities) for Chinese. We are in the process of correcting these cases and anticipate completing this before the public data release.

C Training and Hyperparameters

As discussed in §4, our choices of hyperparameters for both GTT (§C.1) and ITERX (§C.2) follow those associated with the best results in prior work (modulo a change in encoders) and are detailed below. While there is likely room for performance improvements from per-language encoders and hyperparameter tuning, we leave these experiments for future work. The results for these models in the main text are based on single training runs, each of which was conducted on a single 24GB NVIDIA RTX 6000 GPU using the stopping criteria specified below. §C.3 gives details on API hyperparameters and prompts for ChatGPT.

[16] https://en.wikipedia.org/wiki/Cohen%27s_kappa#Interpreting_magnitude
[17] https://cloud.google.com/translate/automl/docs/evaluate#interpretation

C.1 GTT

We use the GTT code base, available here: `https://github.com/xinyadu/gtt`. We use the hyperparameter settings exactly as listed in Appendix B of Du et al. (2021b), with the following changes:

- We used the cased version of mBERT-base (Devlin et al., 2019) as the encoder in lieu of the original uncased BERT-base encoder.
- We train for 30 epochs in all experiments, as we found the default for MUC-4 (18) to be insufficient for convergence in most cases. We use the checkpoint associated with best token-level accuracy on the dev set.

Since the MUC-4 data is uncased, we also experimented with *uncased* mBERT, though we found it yielded consistently worse performance. Devlin et al. (2019) in fact expressly recommend using the cased model, on the grounds that it corrects various issues with the uncased version.[18]

C.2 IterX

We use the ITERX code base, available here: `https://github.com/wanmok/iterx`. We use the same hyperparameters for ITERX as are listed in the "best" column of Table 7 in Chen et al. (2023b), with the following changes:

- We trained on *gold* spans (rather than those predicted by an upstream system), as we empirically found this yielded superior results for MULTIMUC.
- We used mT5-base as the encoder to accommodate all MULTIMUC languages, as discussed in §4.

Chen et al. report only average training time for MUC-4 in their work, but we use the default maximum epochs (150) and patience (30) provided for the MUC-4 training configuration in their repository.

To ensure fair comparison across settings for inference (including for validation), we fix the candidate spans for all three settings to those predicted for the relevant language by the span extraction system of Xia et al. (2021) that we trained for that language in the BI_{MAN} setting (see §4.1).

C.3 ChatGPT

The few-shot experiments described in §4.3 were run using `gpt-3.5-turbo-16k-0613` with a maximum context length of 8,192, a maximum of 1,024 new tokens to be generated, a temperature of 0.5, and a top p of 1.0, with no presence penalty, frequency penalty, or logit biases. A single completion was generated per prompt. We recognize the potential for non-trivial performance variation that may result from even relatively minor changes to a prompt. Given the length of our prompts, cost prohibited us from running multiple variations for the main experiments, so results should be interpreted with caution.

The system prompt for all experiments was as follows:

> You are an expert in information extraction, where you are given a few exemplars to help you understand the task. You have to perform textual analysis on a new document thereafter. Your analysis should be based on the ontology (inferred) and the exemplars.

The structure of the remainder of the prompt is shown below, with prompt-specific components (i.e. the exemplars) described in italicized purple *// comments*. Each "[DOCUMENT TEXT]:" together with the full text document that followed constituted its own **user** message (provided as input in the `messages` API parameter), and each "[TEMPLATES]:" together with the annotated templates that followed likewise constituted its own **assistant** message. The final instructions ("Please follow...") and target document made up the last user message. All templates in the exemplars are formatted in the same way as the one given in the initial instructions below.

[18] See here: `https://github.com/google-research/bert/blob/master/multilingual.md`.

You are given a few exemplars to learn how to perform the template extraction task. You have to learn to do the same extraction to a new document. There are only 5 roles to use: PerpInd, PerpOrg, Target, Victim, Weapon. Valid incident types are: ATTACK, ARSON, ROBBERY, BOMBING, KIDNAPPING, FORCED_WORK_STOPPAGE, BOMBING_OR_ATTACK, ATTACK_OR_BOMBING. A target structures looks like this: Template(incident_type="bombing", PerpInd=[Entity(mentions=[Mention("guerilla column")])], PerpOrg=[Entity(mentions=[Mention("army of national liberation"), Mention("eln")])], Target=[Entity(mentions=[Mention("4-wheel drive vehicle"), Mention("vehicle")])], Victim=[Entity(mentions=[Mention("carlos julio torrado")]), Entity(mentions=[Mention("torrado's son, william"), Mention("william")]), Entity(mentions=[Mention("gustavo jacome quintero")]), Entity(mentions=[Mention("jairo ortega")])], Weapon=[Entity(mentions=[Mention("four explosive charges"), Mention("explosive charges")])])

[EXEMPLARS]:

[DOCUMENT TEXT]:

// full text of example document 1 (least relevant; always in target language)

[TEMPLATES]:

// gold templates for example document 1 (always in target language)

[DOCUMENT TEXT]:

// full text of example document 2 (second most relevant; always in target language)

[TEMPLATES]:

// gold templates for example document 2 (always in target language)

[DOCUMENT TEXT]:

// full text of example document 3 (most relevant; in target language except in BI$_{\text{MAN}}$ *setting)*

[TEMPLATES]:

// gold templates for example document 3 (in target language except in BI$_{\text{MAN}}$ *setting)*

Please follow the previous exemplars to process the new document. You have to use the same domain specific language to describe your extraction results. Do not add additional explanations except for the DSL generated. Make sure that you stick to the exact DSL as shown in the exemplars.

[DOCUMENT TEXT]:

// full text of target (test set) document (always in target language)