TASTY: Interactive Entity Linking As-You-Type

Sebastian Arnold **Robert Dziuba** **Alexander Löser**
Beuth University of Applied Sciences
Luxemburger Straße 10
13353 Berlin, Germany
{sarnold,s58345,aloeser}@beuth-hochschule.de

Abstract

We introduce TASTY (Tag-as-you-type), a novel text editor for interactive entity linking as part of the writing process. Tasty supports the author of a text with complementary information about the mentioned entities shown in a 'live' exploration view. The system is automatically triggered by keystrokes, recognizes mention boundaries and disambiguates the mentioned entities to Wikipedia articles. The author can use seven operators to interact with the editor and refine the results according to his specific intention while writing. Our implementation captures syntactic and semantic context using a robust end-to-end LSTM sequence learner and word embeddings. We demonstrate the applicability of our system in English and German language for encyclopedic or medical text. Tasty is currently being tested in interactive applications for text production, such as scientific research, news editorial, medical anamnesis, help desks and product reviews.

1 Introduction

Entity linking is the task of identifying mentions of named entities in free text and resolving them to their corresponding entries in a structured knowledge base (Hachey et al., 2013). These two steps are often executed as batch process *after* the document has been written by the author. Contrary, doctors during a medical anamnesis, technicians writing supportive manuals or assistants in help desks desire entity linking *during* writing. Ideally, a machine could highlight relevant information about recognized entities while the author is typing the text and gradually adapt the results to complement his task.

Contribution. TASTY is such a novel text editing interface for fine-grained tagging of text articles as part of the writing process. Figure 1 shows an example of the editor in use. While the author is typing characters, a contextual sequence learner immediately recognizes mention boundaries, tags them in-line, resolves associated articles and displays them beside the document. When more context is written, the system reacts and refines boundaries and associations without interrupting the process. The author can *add*, *remove* and *disambiguate* tags according to his task and knowledge. Tasty's extraction model recognizes multi-word mentions and can identify entities that are both in and outside the knowledge base. It does not require linguistic features and is robust to misspelled or out-of-vocabulary words. To our knowledge, Tasty is the first system that implements an interactive entity linking task for manifold scenarios. We apply it to German and English language for encyclopedic and medical text without any change of hyperparameters. In the rest of this paper, we guide through the user interface using a medical examination scenario in Section 2, explain the process of interactive entity linking in Section 3, and conclude in Section 4 with an evaluation and discussion. A live demo and video of Tasty can be found at http://dbl43.beuth-hochschule.de/demo/tasty/

2 Demonstration Scenario

TASTY supplies doctors with supplemental materials. As demonstration example we showcase a medical *History and Physical Examination (H&P)* write-up, where doctors write text about a patient's

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations, pages 111–115, Osaka, Japan, December 11-17 2016.

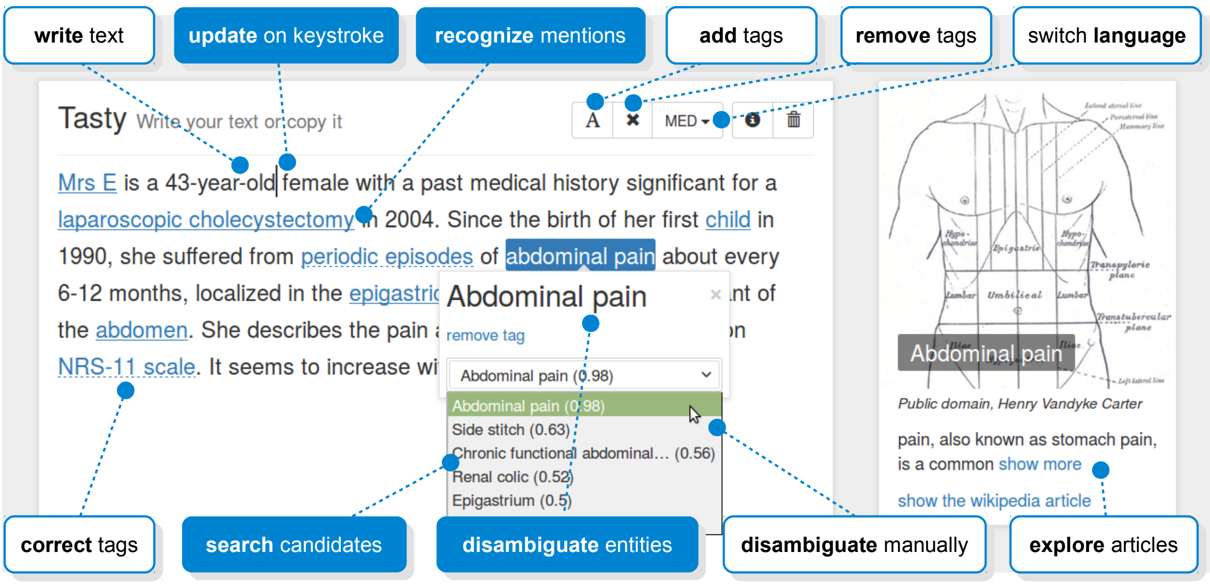

Figure 1: Example of writing a text in Tasty's user interface. Named entities are displayed as tags, articles appear on the right side. White boxes denote interaction operators, filled boxes show system actions.

history and conditions. Tasty can recognize these medical conditions and link them to Wikipedia articles. Other possible targets are e.g. research papers or relevant archived doctor letters. As a result, a doctor may learn from these documents additional insights for sharpening her focus in the write-up.

We showcase the following scenario as an example H&P (see Figure 1): The doctor starts by writing the first sentence about her patient: "Mrs E is a 43-year-old female with a past medical history significant for a laparoscopic cholecystectomy". Tasty responds to key strokes, recognizes mentions, searches for candidates and displays a complementary article for *cholecystectomy* next to the document. The doctor might explore the article and incrementally learn about important aspects of this condition. She might continue writing "she suffered from periodic episodes of abdominal pain localized in the epigastric region" and manually select a more precise disambiguation for the phrase *abdominal pain*. She may correct further tagging errors, e.g. remove the unwanted tag *Mrs E*. In case of a missing tag, the doctor can edit a phrase, e.g. NRS-11 pain scale and tag it manually. The system reacts and returns a disambiguation.

3 Interactive Entity Linking Process

We implement interactive entity linking using *mention recognizer*, *candidate searcher* and *link disambiguator* stages (Hachey et al., 2013). We extend the process by an interactive cycle that includes *partial update* and *user feedback* operators, as shown in Figure 2. We implement Tasty as demonstrator for English (EN) and German language (DE) and a specialized medical scenario (MED).

Step 1: Update while the author is typing. Tasty's user interface is based on a lightweight rich text editor[1] that we extend to display named entity mentions as in-line tags. Tasty captures the author's key strokes and detects word boundaries after space or punctuation characters. We split a document of length n into a sequence of word tokens $d = (w_1, \ldots, w_n)$ using a language-independent whitespace tokenizer[2]. In a partial update step, we analyze only the changed portion $\tilde{d} = (w_b, \ldots, w_e)$, $1 \leq b < e \leq n$ of the document. We expand indexes b and e to sentence boundaries and omit any further linguistic processing.

Step 2: Recognize mention boundaries. We define a mention m as the longest possible span of adjacent tokens that refers to a an entity or relevant concept of a real-world object, such as epigastric region. In Tasty, we further assume that mentions are non-recursive and non-overlapping. The objective of this step is to detect all mention spans $M_{\tilde{d}} = \{m_i\}$ in the document portion. We model this task as

[1] We use Quill v1.0.0-beta.11 http://quilljs.com

[2] We use PTBTokenizer from Stanford CoreNLP 3.6.0 http://stanfordnlp.github.io/CoreNLP/

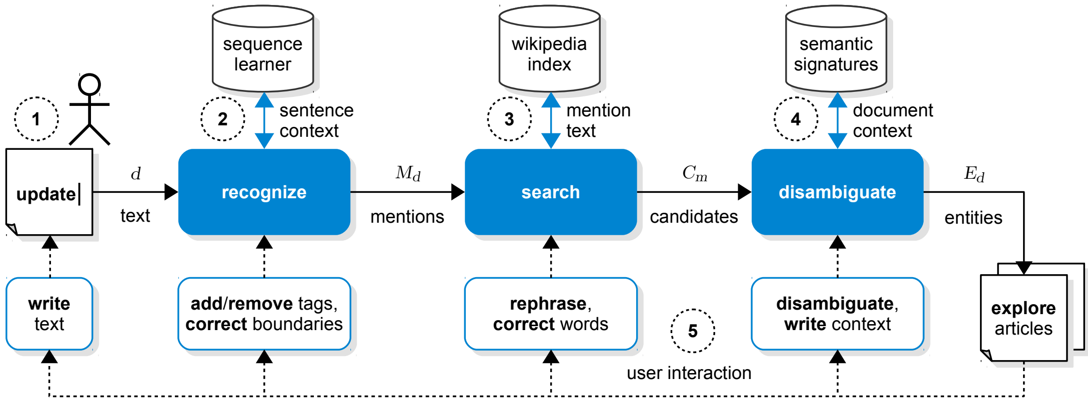

Figure 2: Overview of the interactive entity linking process in Tasty. While the author is writing a text, the system recognizes mentions, searches for entity candidates and disambiguates the mention to its corresponding Wikipedia article. The author is able to interact with every stage of the extraction process.

context-sensitive sequential word labeling problem. We predict for each token $w_t \in \tilde{d}$ a target label $\hat{y}_t$ according to the BIOES tagging scheme (Ratinov and Roth, 2009) with respect to its surrounding words (Eq. 1). From these labels, we populate $M_{\tilde{d}}$ in a single iteration. For the prediction task, we utilize long short-term memory (LSTM) networks (Hochreiter and Schmidhuber, 1997), which are are able to capture long-range sequential context information with short answer times. The input is a sequence of word feature vectors $x(w_t)$ with three components: First, we use lowercase letter-trigram word hashing (Huang et al., 2013) to encode word syntax on character level. This technique splits a word into discriminative three-letter 'syllables' with boundary markers, e.g. cell $\rightarrow$ {#ce, cel, ell, ll#} to make the bag robust against misspellings and out-of-vocabulary words. Second, we utilize word embeddings (Mikolov et al., 2013)[3] to represent word semantics in dense vector space. Third, we encode surface form features by generating a vector of flags that indicate e.g. initial capitalization, uppercase, lower case or mixed case.

$$\hat{y}_t = \underset{l \in \{B,I,O,E,S\}}{\arg\max} \; \mathrm{p}\big(y_t = l \mid x(w_b), \ldots, x(w_{t-1}), x(w_t), x(w_{t+1}), \ldots, x(w_e)\big) \tag{1}$$

We pass through $\tilde{d}$ bidirectionally using a stacked BLSTM+LSTM architecture (Arnold et al., 2016)[4]. Our recognition component can be trained 'end-to-end' with only few thousand labeled sentences. For the demonstration, we provide three different pre-trained models: EN is trained to recognize named entities (persons, organizations, locations and misc) in English encyclopedic text, DE captures proper nouns (untyped) in German encyclopedic text, and MED recognizes biomedical terms in scientific text.

Step 3: Search for candidate links. Our next step is to resolve a subset of Wikipedia article candidates C_m for each of the detected mentions m. We especially aim to capture a large number of candidates for highly ambiguous terms such as scale or child. For this task, we create an index of 4.5M English and 1.6M German Wikipedia abstracts[5]. We use redirects and anchor phrases to capture alternative writings and synonyms (Hachey et al., 2013). We apply a dictionary-based technique described by Ling et al. (2015) and query the index for candidates $C_m = \{c_j \mid \forall m \in \tilde{d} : c.\text{title} \approx m.\text{span}\}$ using phrase queries with BM25 similarity[6] for retrieval. In case of an empty result, we return NIL (non-linkable entity).

Step 4: Disambiguate associated articles. From the set of candidates C_m, we want to pick the most likely entity associations $E_d = \{(m_i, \hat{c}_j)\}$. We do this by picking the candidate $\hat{c}$ with maximum score depending on the mention and current document context (Eq. 2). As scoring function, we utilize short text similarity (Kenter and de Rijke, 2015) between mention context $m.d$ and a candidate article $c.d$. We

[3]We trained a 150-dimensional lowercase word2vec model using English and German CoNLL2003 and Wikipedia articles
[4]We implement the network using Deeplearning4j 0.6.0 with CUDA backend `https://deeplearning4j.org`
[5]We use DBpedia version 2015-10 `http://wiki.dbpedia.org/datasets`
[6]We use the implementation in Lucene 6.1.0 `http://lucene.apache.org`

stage	**Recognize (EN)**			**Recognize (DE)**			**Recognize (MED)**			**Disambiguate (EN)**		
dataset	CoNLL2003 NER			TIGER Treebank			GENIA Corpus			DBpedia Spotlight		
corpus	Reuters RCV-1			Fr. Rundschau			Medline abstracts			Wikipedia		
language	English			German			English			English		
domain	newswire			newswire			biomedical			encyclopedia		
annotation	named entities			proper nouns			medical terms			Wikipedia IDs		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
Stanford NER	**96.4**	73.6	83.5	68.9	31.7	43.4	31.7	7.6	12.3	–	–	–
Lingpipe	69.0	50.3	58.2	–	–	–	**91.8**	**93.8**	**92.8**	–	–	–
DBpedia Spotlight	66.6	58.6	62.4	–	–	–	–	–	–	**82.0**	62.1	**70.7**
Babelfy	44.2	62.7	51.8	–	–	–	–	–	–	57.7	46.7	51.6
TASTY	90.3	**92.0**	**91.1**	**82.7**	**83.9**	**83.3**	77.5	79.5	78.5	66.1	**64.9**	65.4

Table 1: Evaluation of Tasty's recognition and disambiguation stages (micro-averaged exact span match).

utilize word embeddings to calculate vectors $v(w_t)$ for every token in the document and aggregate them into a normalized mean document vector that we use as semantic signature $s(d)$ (Eq. 3). We finally use cosine similarity between the semantic signatures as scoring function (Eq. 4).

$$\hat{c} = \arg\max_{c \in C_m} \mathrm{score}(c|m,d) \quad (2) \qquad s(d) = \frac{1}{n} \sum_{w_t \in d} v(w_t) \quad (3) \qquad \mathrm{score}(c|m,d) = \frac{s(m.d) \cdot s(c.d)}{\|s(m.d)\| \|s(c.d)\|} \quad (4)$$

Step 5: Feed back user interaction. Tasty offers seven feedback operators that enable an author to interact with every component in the extraction process. All operators are based on typing or text selection. Using **write**, the author emits more context and the system reacts to word boundaries by triggering a partial update. The author might also **rephrase** single words, triggering the system to update surrounding annotations. Using the **add** button, the author is able to correct false negative predictions from the recognition component. The system will tag the selected mention, generate candidates and decide for an associated article. The **remove** button deletes selected tags to correct false positive predictions. The author can **correct** boundaries of an existing tag, and the system will update the link if necessary. If the boundaries of a tag are correct, but the link is not, the author can **disambiguate** by assigning a different candidate from the drop-down menu. Finally, the author benefits from several operators to **explore** the articles. Corrections are directly executed in the local session and fed back as training data to our model.

4 Evaluation

We evaluate Tasty's recognition and disambiguation stages compared to four state-of-the-art annotators: Stanford NER[7] and LingPipe[8] implement text chunking classifiers with pre-trained models. DBpedia Spotlight (Mendes et al., 2011) and Babelfy (Moro et al., 2014) are comprehensive systems specialized for entity linking and word sense disambiguation. We run the experiments in an isolated offline setting using the GERBIL evaluation framework (Usbeck et al., 2015) and measure micro-averaged precision, recall and NER-style F1 score for exact span match. For the recognition stage, we use test splits from English CoNLL-2003 shared task (Tjong Kim Sang and De Meulder, 2003), German TIGER Corpus (Brants et al., 2004) and biomedical GENIA Corpus (Ohta et al., 2002) datasets. For the disambiguation stage, we utilize the English DBpedia Spotlight NIF NER Corpus (Mendes et al., 2011).

Result discussion. Table 1 shows the evaluation results. We notice that Tasty's recognition stage is able to adapt to English (91.1% F1) and German newswire (83.3% F1) and English biomedical texts (78.5% F1) using small training sets of only 4000 labeled sentences and without any change of hyperparameters. This result for 'raw' mention recognition is on par with state-of-the-art text chunkers (Arnold et al., 2016) and achieves significantly higher recall on news datasets. The fact that we cannot achieve best results on biomedical text is due to generalization: while the pre-trained LingPipe model is strongly overfitted to GENIA dictionaries, Tasty leverages context and typical syllables and therefore is able to

[7] We use English CoNLL 4-class distsim CRF and German dewac CRF models `http://nlp.stanford.edu/software/CRF-NER.shtml`

[8] We use MUC6 CharLmRescoringChunker and GENIA TokenShapeChunker `http://alias-i.com/lingpipe/`

scenario	**Research**	**Editorial**	**Diagnosis**	**Help Desk**	**Shopping**
example	report writing	news authoring	anamnesis	customer support	product order
subtasks	pin topics find sources lookup explain	annotate paragraphs identify topics and tags thesaurus style suggestion	lexicon search patient history side effects medical compatibility	FAQ search related tickets manuals expertise search	price comparison feature infobox user reviews purchase advice

Table 2: Examples of Tasty's application in five scenarios and potential exploratory subtasks.

detect mention boundaries even if the word is misspelled or not priorly known to the system, e.g. "we treat the XYDKF34 cells with high-dosed srscklartamin." Furthermore, Tasty's disambiguation stage shows comparable performance to the comprehensive systems on the English disambiguation task (65.4% F1).

Applying TASTY. We showcased Tasty's editor with pre-trained models to 21 experienced professionals and learned about exciting application scenarios which are shown in Table 2. A large group of users applied the results of in-line entity linking to subtasks with *exploratory search intention* (Marchionini, 2006): *look up* facts or definitions for entities in the text, *learn* from complementary articles, *compare* written text against text in archives, *verify* information, *integrate* with existing tagging schemes. For future implementations, users suggested the application of *investigatory* subtasks: *evaluate* text to fit a desired tone or vocabulary, *discover* alternatives or get *advice* from user reviews or experts. For realizing these application scenarios, in our future work we will extend Tasty with powerful cross-document coreference capabilities and specialized retrieval models for a broader set of data sources.

Acknowledgements Our work is funded by the Federal Ministry of Economic Affairs and Energy (BMWi) under grant agreement 01MD15010B (Project: Smart Data Web).

References

Sebastian Arnold, Felix A. Gers, Torsten Kilias, and Alexander Löser. 2016. Robust Named Entity Recognition in Idiosyncratic Domains. In *arXiv:1608.06757 [cs.CL]*.

Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther König, et al. 2004. TIGER: Linguistic Interpretation of a German Corpus. *Research on Language and Computation*, 2(4):597–620.

Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R. Curran. 2013. Evaluating Entity Linking with Wikipedia. *Artificial intelligence*, 194:130–150.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.*, 9(8):1735–1780.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning Deep Structured Semantic Models for Web Search using Clickthrough Data. In *CIKM'13*, pages 2333–2338. ACM.

Tom Kenter and Maarten de Rijke. 2015. Short Text Similarity with Word Embeddings. In *CIKM'15*, volume 15, page 115. ACM.

Xiao Ling, Sameer Singh, and Daniel S Weld. 2015. Design Challenges for Entity Linking. *ACL'15*, 3:315–328.

Gary Marchionini. 2006. Exploratory Search: From Finding to Understanding. *CACM*, 49(4):41–46.

Pablo N Mendes, Max Jakob, Andrés Garcia-Silva, and Christian Bizer. 2011. DBpedia Spotlight: Shedding Light on the Web of Documents. In *I-Semantics 2011*, pages 1–8. ACM.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. *arXiv:1301.3781 [cs.CL]*.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity Linking meets Word Sense Disambiguation: A Unified Approach. *ACL'14*, 2:231–244.

Tomoko Ohta, Yuka Tateisi, and Jin-Dong Kim. 2002. The GENIA Corpus: An Annotated Research Abstract Corpus in Molecular Biology Domain. In *HLT'02*, pages 82–86. Morgan Kaufmann.

Lev Ratinov and Dan Roth. 2009. Design Challenges and Misconceptions in Named Entity Recognition. In *CoNLL'09*, pages 147–155. ACL.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *CoNLL'03*, pages 142–147. ACL.

Ricardo Usbeck, Michael Röder, Axel-Cyrille Ngonga Ngomo, Ciro Baron, Andreas Both, et al. 2015. GERBIL: General Entity Annotator Benchmarking Framework. In *WWW'15*, pages 1133–1143. IW3C2.