SUFT-1, a system for helping understand spontaneous multilingual and code-switching tweets in foreign languages : experimentation and evaluation on Indian and Japanese tweets

Ritesh Shah

Communauté UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES

Spécialité : Informatique

Arrêté ministériel : 25 mai 2016

Présentée par

Ritesh SHAH

Thèse dirigée par **Christian BOITET**, retraité
et codirigée par **Pushpak BHATTACHARYYA**
préparée au sein du **Laboratoire d'Informatique de Grenoble**
dans **l'École Doctorale Mathématiques, Sciences et technologies de l'information, Informatique**

SUFT-1, un système pour aider à comprendre les tweets spontanés multilingues et à commutation de code en langues étrangères : expérimentation et évaluation sur les tweets indiens et japonais

SUFT-1, a system for helping understand spontaneous multilingual and code-switching tweets in foreign languages: experimentation and evaluation on Indian and Japanese tweets

Thèse soutenue publiquement le **27 octobre 2017**,
devant le jury composé de :

Monsieur GEORGES ANTONIADIS
PROFESSEUR, UNIVERSITE GRENOBLE ALPES, Président
Monsieur PATRICK PAROUBEK
INGENIEUR DE RECHERCHE, CNRS DELEGATION ILE-DE-FRANCE SUD, Rapporteur
Monsieur MATHIEU LAFOURCADE
MAITRE DE CONFERENCES, UNIVERSITE DE MONTPELLIER, Rapporteur
Madame VIOLAINE PRINCE
PROFESSEUR, UNIVERSITE DE MONTPELLIER, Examinateur
Monsieur CLEMENT LEVALLOIS
MAITRE DE CONFERENCES, EM-LYON BUSINESS SCHOOL, Examinateur
Monsieur Mathieu MANGEOT
MAITRE DE CONFERENCES, UNIVERSITE SAVOIE MONT BLANC, Encadrant

Abstract

As TWITTER evolves into a ubiquitous information dissemination tool, understanding tweets in foreign languages becomes an important and difficult problem. Because of the inherent code-mixed[1], disfluent and noisy nature of tweets, state-of-the-art Machine Translation (MT) is not a viable option (Farzindar & Inkpen, 2015). Indeed, at least for Hindi and Japanese, we observe that the percentage of "understandable" tweets falls from 80% for natives to below 30% for target (English or French) readers using GOOGLE TRANSLATE or YANDEX. Our starting hypothesis is that it should be possible to build generic tools, which would enable foreigners to make sense of at least 70% of "native tweets", using a versatile "active reading" (AR) interface, while simultaneously determining the percentage of understandable tweets under which such a system would be deemed useless by intended users.

We have thus specified a generic "SUFT" (System for helping Understand Foreign Tweets), and implemented SUFT-1, an interactive multi-layout system based on AR, and easily configurable by adding dictionaries, morphological modules, and MT plugins. It is capable of accessing multiple dictionaries for each source language and provides an evaluation interface. For evaluations, we introduce a task-related measure inducing a negligible cost, and a methodology aimed at enabling a « continuous evaluation on open data », as opposed to classical measures based on test sets related to closed learning sets. We propose to combine *understandability ratio* and *understandability decision time* as a two-pronged quality measure, one subjective and the other objective, and experimentally ascertain that a dictionary-based active reading presentation can indeed help understand tweets better than available MT systems.

In addition to gathering various lexical resources, we constructed a large resource of "word forms" appearing in Indian tweets with their morphological analyses (163221 Hindi word forms from 68788 lemmas and 72312 Marathi word forms from 6026 lemmas) for creating a multilingual morphological analyzer specialized to tweets, which can handle code-mixed tweets, compute unified features, and present a tweet with an attached AR graph from which foreign readers can intuitively extract a plausible meaning, if any.

Résumé

Alors que TWITTER évolue vers un outil omniprésent de diffusion de l'information, la compréhension des tweets en langues étrangères devient un problème important et difficile. En raison de la nature intrinsèquement à commutation de code, discrète et bruitée des tweets, la traduction automatique (MT) à l'état de l'art n'est pas une option viable (Farzindar & Inkpen, 2015). En effet, au moins pour le hindi et le japonais, nous observons que le pourcentage de tweets « compréhensibles » passe de 80% pour les locuteurs natifs à moins de 30% pour les lecteurs en langue cible (anglais ou français) utilisant GOOGLE TRANSLATE ou YANDEX. Notre hypothèse de départ est qu'il devrait être possible de créer des outils génériques, permettant aux étrangers de comprendre au moins 70% des « tweets locaux », en utilisant une interface polyvalente de « lecture active » (LA, AR en anglais) tout en déterminant simultanément le pourcentage de tweets compréhensibles en-dessous duquel un tel système serait jugé inutile par les utilisateurs prévus.

Nous avons donc spécifié un « SUFT » (système d'aide à la compréhension des tweets étrangers) générique, et mis en œuvre SUFT-1, un système interactif à présentation multiple basé sur la LA, et facilement configurable en ajoutant des dictionnaires, des modules morphologiques et des plugins de TA. Il est capable d'accéder à plusieurs dictionnaires pour chaque langue source et fournit une interface d'évaluation. Pour les évaluations, nous introduisons une mesure liée à la tâche induisant un coût négligeable, et une méthodologie visant à permettre une « évaluation continue sur des données ouvertes », par opposition aux mesures classiques basées sur des jeux de test liés à des ensembles d'apprentissage fermés. Nous proposons de combiner le *taux de compréhensibilité* et le *temps de décision de compréhensibilité* comme une mesure de qualité à deux volets, subjectif et objectif, et de vérifier expérimentalement qu'une présentation de type lecture active, basée sur un dictionnaire, peut effectivement aider à comprendre les tweets mieux que les systèmes de TA disponibles.

En plus de rassembler diverses ressources lexicales, nous avons construit une grande ressource de "formes de mots" apparaissant dans les tweets indiens, avec leurs analyses morphologiques (163221 formes de mots hindi dérivées de 68788 lemmes et 72312 formes de mots marathi dérivées de 6026 lemmes) pour créer un analyseur morphologique multilingue spécialisé pour les tweets, capable de gérer des tweets à commutation de code, de calculer des traits unifiés, et de présenter un tweet en lui attachant un graphe de LA à partir duquel des lecteurs étrangers peuvent extraire intuitivement une signification plausible, s'il y en a une.

[1] See Definition 3.

Abstract in Hindi

ट्विटर के रूप में एक सर्वव्यापी सूचना प्रसार उपकरण के विकसित होते ही विदेशी भाषाओं में ट्वीट्स को समझने की समस्या एक महत्वपूर्ण और कठिन चुनौती बनके सामने आती है। ट्वीट्स में निहित कोड-मिक्सिंग, विसंगत वाक्य रचना एवं सामान्यतः अशुद्ध लेखन की वजह से, अत्याधुनिक मशीन ट्रांसलेशन (एमटी) एक व्यवहार्य विकल्प नहीं है (फारज़ींदर एंड इंकपेन, 2015)। वास्तव में, कम से कम हिंदी और जापानी के लिए, हम देखते हैं कि गूगल ट्रांस्लेट या यैंडेक का उपयोग करते हुए, "समझने योग्य" ट्वीट्स का प्रतिशत किसी मूल निवासी के लिए 80% से गिरकर किसी अंग्रेज़ी या फ्रेंच वाचक के लिए 30% हो जाता है। हमारी प्रारंभिक अवधारणा यह है कि एक बहुमुखी "एक्टिव रीडिंग" (एआर) इंटरफ़ेस का उपयोग करते हुए विदेशियों को कम से कम 70% "देशी ट्वीट्स" का अर्थ समझने में सक्षम कर सके ऐसे एक व्यापक उपकरण बनाने की निश्चित रूप से संभावना है। साथ ही साथ हम ये भी सुनिश्चित करते है कि कम से कम कितने प्रतिशत ट्वीट्स न समझ आने पर ये उपकरण प्रयोक्ताओं द्वारा बेकार माना जाएगा।

इस अवधारणा के आधार पर हमने एक व्यापक "एसयूएफटी" (विदेशी ट्वीट्स को समझने में मदद करनेवाला सिस्टम) निर्दिष्ट किया, एवं तत्पश्चात "एक्टिव रीडिंग" पर आधारित एक इंटरैक्टिव मल्टी-लेआउट सिस्टम(उपकरण) SUFT-1 का कार्यान्वन किया। इस उपकरण का प्रारूप आसानी से शब्दकोश, रूपिकी या शब्द साधन मॉड्यूल और मशीनी अनुवाद के प्लगइन्स जोड़कर बदला जा सकता है। यह प्रत्येक भाषा के लिए एकाधिक शब्दकोशों का उपयोग करने एवं एक मूल्यांकन इंटरफ़ेस प्रदान करने में सक्षम है। मूल्यांकन के लिए, हम एक कार्य-संबंधित माप और एक कार्यप्रणाली का प्रस्ताव रखते हैं जो नगण्य लागत से "ओपन डाटा पर निरंतर मूल्यांकन" करने में सक्षम है एवं उन शास्त्रीय उपायों से अलग है जो "क्लोस्ड लर्निंग सेट्स" पर आधारित हैं।
हम 'अंडरस्टँडेबिलिटी रेशियो' एवं 'अंडरस्टँडेबिलिटी डिसीज़न टाइम' को व्यक्तिपरक और वस्तुपरक माप की दृष्टी से दो-तरफा गुणवत्ता वाले एक माप के रूप में जोड़ते हैं। साथ ही साथ प्रयोगात्मक रूप से यह पता लगाते है कि क्या एक शब्दकोश-आधारित सक्रिय रीडिंग प्रस्तुति वास्तव में उपलब्ध एमटी सिस्टमों की अपेक्षा ट्वीट्स को बेहतर समझने में सहायक हो सकती है। विभिन्न शब्दार्थिक संसाधनों को इकट्ठा करने के अलावा, हमने भारतीय ट्वीट्स में निहीत "वर्ड फॉर्म्स" का उनके रूपात्मक विश्लेषण के साथ एक बड़ा संसाधन निर्मित किया है जिसमें (68788 लेम्माज़ से 163221 हिंदी वर्ड फॉर्म्स और 6026 लेम्माज़ से 72312 मराठी वर्ड फॉर्म्स) हैं। यह एक बहुभाषी रूपात्मक विश्लेषक बनाने के लिए है, जो कि कोड-मिश्रित ट्वीट्स को संसाधित कर सकता है, एकीकृत वैशिष्ट्यों की गणना कर सकता है और एक्टिव रीडिंग ग्राफ के साथ एक ट्वीट प्रस्तुत कर सकता है जिससे विदेशी पाठक सहजता से संभाव्य अर्थ निकाल सकें।

Abstract in Marathi

ट्विटरच्या रूपात एका सर्वव्यापी माहिती प्रसार उपकरणचा विकास झाल्याबरोबर परदेशी भाषांमध्ये ट्वीट्स समजून घेण्याचे एका महत्त्वाचे आणि कठीण आव्हान समोर उभे राहते. ट्वीट्स मध्ये समाविष्ट कोड-मिक्सिंग, विसंगत वाक्य रचना आणि सहसा अशुद्ध लेखन यांच्यामुळे अत्याधुनिक मशीन ट्रांसलेशन (एमटी) एक व्यवहार्य विकल्प म्हणून येत नाही (फारजींदर एंड इंकपेन, 2015). खरं तर, हिंदी आणि जपानी भाषांसाठीतरी आपण बघतो कि गूगल ट्रांस्लेट किंवा यैंडेकचा वापर करताना, "समजण्यासारखे" ट्वीट्स यांची टक्केवारी मूळ निवाश्यांसाठी 80% पासून कुणा इंग्रजी किंवा फ्रेंच वाचकसाठी 30% होउन जाते. आमची प्रारंभिक संकल्पना अशी आहे कि एक अष्टपैलू "ॲक्टिव रीडिंग" (एआर) इंटरफेसचा वापर करताना परदेश्यांना किमान 70% "देशी ट्वीट्स", यांचा अर्थ समझण्यास सक्षम करू शकेल असे एक व्यापक उपकरण बनवण्याची निश्चित शक्यता आहे. त्याच वेळी, आम्ही हे देखील सुनिश्चित करतो की अमुक टक्के ट्वीट्स अर्थहीन ठरल्यास हे उपकरण वापरकर्त्यांद्वारे निरुपयोगी ठरेल.

या संकल्पनांवर आधारित, आम्ही एक व्यापक "एसयूएफटी" (परदेशी ट्वीट्स समझण्यास मदत करणारा सिस्टम) निर्दिष्ट केला, आणि त्यानंतर "सक्रिय वाचन" यावर आधारित एक इंटरॅक्टिव मल्टी-लेआउट सिस्टम(उपकरण) SUFT-1 लागू केला. या साधनाचे स्वरूप सहजपणे शब्दकोश, आवृत्ती किंवा शब्द साधन मॉड्यूल आणि मशीनी अनुवादचे प्लगइन्स जोडून बदलले जाऊ शकतात. हे प्रत्येक भाषेसाठी एकाधिक शब्दकोश वापरण्यास आणि एक मूल्यमापन इंटरफेस प्रदान करण्यास सक्षम आहे. मूल्यमापनसाठी आम्ही एका कामा संबन्धित मोजमापाचा आणि एका पद्धतीचा प्रस्ताव ठेवतो जे कमीतकमी खर्च्यात "ओपन डाटा वर सतत मूल्यमापन" करण्यास सक्षम आहे आणि "क्लोस्ड लर्निंग सेट्स" वर आधारित असलेल्या शास्त्रीय उपायांपेक्षा वेगळे आहे.

आम्ही 'अंडरस्टँडेबिलिटी रेशियो' आणि 'अंडरस्टँडेबिलिटी डिसीजन टाइम' ला व्यक्तिनिष्ठ आणि उद्दिष्ट मोजणीच्या दृष्टीने एका दोन-मार्गी गुणवत्ता म्हणून वापरतो. तसेच प्रायोगिक दृष्ट्या आम्ही हे जाणून घेतो कि खरोखर शब्दकोश-आधारित सक्रिय वाचन सादरीकरण ट्वीट्सना समझण्यासाठी उपलब्ध एमटी सिस्टम्सपेक्षा चांगले होउ शकतात का? विविध शब्दार्थासंबंधीची संसाधने गोळा करण्या व्यतिरिक्त आम्ही भारतीय ट्वीट्स मध्ये समाविष्ट "वर्ड फॉर्म्स" चा त्यांच्या रूपात्मक विश्लेषणासह एक चांगला स्रोत तयार केला आहे ज्या मध्ये (68788 लेम्माज यांपासून 163221 हिंदी वर्ड फॉर्म्स आणि 6026 लेम्माज यांपासून 72312 मराठी वर्ड फॉर्म्स) आहेत. हे एक बहुभाषी शब्दविज्ञान विश्लेषक तयार करण्यासाठी आहे जे कोड मिश्रित ट्वीट्स हाताळू शकते, एकात्मिक वैशिष्ट्ये मोजू शकते आणि सक्रिय वाचन ग्राफसह एक ट्वीट सादर करू शकते जेणेकरून परदेशी वाचक सहजतेने ट्वीट्सचे शक्य ते अर्थ काढू शकतात..

In memoriam

To my dad

Shri. Mahendra Rupchand Shah
(1951-2009)

Acknowledgments & thanks

Firstly, I would like to express my sincere gratitude to my director Prof. Christian Boitet for arranging this doctorate for me, for his patience, motivation, and immense knowledge. His invaluable guidance helped me throughout the research and writing of this thesis. It has been a privilege to be associated with him and draw from over 4 decades of his experience in Computational Linguistics. His devotion towards scientific work, attention to details and a constant endeavour to push oneself will always be a motivation for me.

I am also sincerely grateful to my co-director Prof. Pushpak Bhattacharyya with whom I have been associated since before my Ph.D. I will always remain thankful to him for his scientific guidance and encouragement. I have learnt immensely by watching him spearhead many a NLP missions in and for India. I thank him for always allowing me to be a part of the CFILT team at IITB and entrusting me with many responsibilities.

Again, I express a deep sense of gratitude towards my teachers and, in keeping with the '*guru-shishya parampara*', the teacher-student tradition of India, resolve to assimilate all the teachings and make utmost attempts to take them forward.

I would like to thank my co-encadrant Dr. Mathieu Mangeot who has been always helpful with his scientific suggestions and constant moral support.

I would also like to thank the other members of my thesis jury: Prof. Georges Antoniadis, Dr. Patrick Paroubek, Dr. Mathieu Lafourcade, Prof. Violaine Prince and Dr. Clément Levallois, for their participation, insightful comments and encouragements.

My sincere thanks also go to Prof. Akiko Aizawa and Dr. Asanobu Kitamoto, who accepted me as a research intern at the NII labs, Tokyo.

I would like to thank the MSTII doctoral school, UJF, IRD and then UGA and LIG (France), IITB (India) and NII (Tokyo). These institutions allowed me to pursue my studies and research with their support. During my doctorate, I was confronted with several cultural and linguistic changes and I take this opportunity then, to acknowledge all my friends and colleagues of the respective labs who have made my journey, an endearing experience.

I extend a special thanks to Dr. Georges Fafiotte who was the first colleague and friend to receive me in Grenoble, and helped me with practically everything in Grenoble.

From IITB (CFILT labs) in India, I thank Anand, Anoop, Samir, Bala, Abhijit, Rajen, Diptesh and, all my co-authors & friends for the collective work and stimulating discussions.

From NII labs in Tokyo, I would like to thank Goran for the linguistic exchanges and crucial programming help. I also thank Lucas, Inoe, Binyam and Fumiki for participating in my evaluation studies.

From LIG lab in Grenoble, I thank my friends and fellow *doctorants* Ruslan, Elina, Houssein, and post-docs Andon, Claire, Lingxiao, Ying, and will always cherish their friendship.

I also thank the senior members of LIG: Valérie Bellynck, Didier Schwab, Laurent Besacier, Jean-Philippe Guilbaud, Jean-Claude Durand and Mutsuko Tomokiyo, for their counsel and co-operation in so many ways.

Most importantly, I owe an acknowledgment to my family who made this doctorate possible. I owe my deepest gratitude to my parents Shri. Mahendra Shah and Smt. Saroj Shah for their unconditional love and faith in me. I also thank my brother Akhil together with my sister-in-law Srushti for his considerate backing and his quiet concern for me at all times.

Last but not least, I must mention that I especially missed seeing my little daughter Rushali grow up during the past few years and I thank my wife Ruchika for taking care of her, for standing by me and selflessly encouraging me to take up this doctorate away from home.

Table of contents

List of figures

List of tables

Typographical conventions

1. Throughout this thesis, citations will be indicated by a special character style and a special paragraph style. For example,

   Bridge enables rapid translation of social media

2. Systems or applications or format names are in a special character style. For example,
   GOOGLE TRANSLATE, YANDEX, HTML

3. Computer programs, data and messages use another character style. For example,
   `get_senses_from_onlineDict(tweetWords, dictionaryName)`

Glossary

API	Application Programming Interface
AR	Active reading
AR+MT_based	Active reading and MT-based
ATEF	Analyse de Textes en États Finis (Finite State Text Analysis), a specialized language to write morphological analyzers of the Ariane-G5 MT platform, based on an extended non-deterministic finite state string transducer model.
EBMT	Example-Based Machine Translation (another type of empirical MT)
EMT	Expert MT (based on linguistic knowledge, including internal semantics)
HUFTweets	Help for Understanding Foreign Tweets
IAST	International Alphabet of Sanskrit Transliteration
KBMT	Expert MT (Knowledge-based MT, based on linguistic and external semantic knowledge)
MA	Morphological Analysis
ML	Machine Learning
MT	Machine Translation
NE	Named Entity
NLP	Natural Language Processing
OOV	Out Of Vocabulary [words]
REST	REpresentational State Transfer
SL	Source Language
TL	Target Language
SMT	Statistical Machine Translation
SUFT	System for helping Understand Foreign Tweets
SUFT-1	Version 1 of our system for HUFTweets
UFT	Understanding Tweets in Foreign Languages (in general)
UFTweets	Understanding foreign tweets (in general)
UNREC	Unrecognized [words], i.e. words that are OOV and are also not recognized as derivatives or compounds
UNL	Universal Networking Language
WSD	Word Sense Disambiguation

Introduction

This thesis is at the crossroads between « NLP for social data » (Farzindar & Inkpen, 2015), « multilingual computing », « Indian Language Technology », and « semantic Web ». From the beginning, it has been set in the framework of a bilateral scientific cooperation between France and India[2], and also got the support of NII[3] in Japan. After 8 years of collaborative work in NLP on Indian languages and English (in Machine Translation as well as on IndoWordnet and, corpus processing), we formulated and submitted[4] a project to CEFIPRA[5], called BigTextIF, aiming at building a recommender system for tourists in India, based on local tweets. Our thesis was to be dedicated to that project and supposed the collaboration of many researchers and master students from several Indian and French research groups[6]. Unfortunately, BigTextIF was not funded, so that we had to adapt our thesis, from project-oriented to tool-oriented, with less ambitious goals, while at the same time determining interesting scientific topics. Preliminary investigations done while preparing the BigTextIF project had shown that processing Indian tweets *per se* was quite interesting and challenging, because of their *big data* aspect[7] and of their intrinsic difficulty[8]. We therefore decided to concentrate on their understanding by foreign visitors. As we were fortunate to be invited to NII for 2 international internships, we got the opportunity to also work on Japanese tweets, and enlarge our goal to the study of tools and methods to help understand tweets in foreign languages (and not only in Indian languages). We could then also take the role of a visitor trying to understand local tweets without knowing the language at all.

Research on NLP for Indian languages has been very active since at least 1980. It started in Kanpur (under the direction of Prof. R.M.K. Sinha), and soon there were national projects for computerizing the 22 official[9] languages (using almost as many different scripts), and then to produce tools such as document processors including data entry methods, spellcheckers, hyphenators, etc. A national project aiming at producing Machine Translation for 8 languages has been going on since 1995 or 2000[10]. Since 2000, many NLP centers have been created. In 1996, under the direction of Prof. Pushpak Bhattacharyya, the CFILT[11] of IITB[12] joined the international UNL project. In 2002, CFILT organized a very successful UNL symposium in Goa. It also organized the IndoWordnet project[13], and many subsequent conferences in India, notably ONI-2008 (on ontologies and Wordnets), ICON conferences, and in 2012 one of the largest and most famous NLP conferences, COLING, in Mumbai.

When preparing the BigTextIF project proposal, one main goal was to work on *big data*, as it was one essential topic in the Call for Projects. We were quite interested in that aspect, having

[2] IITB in India, UJF (then UGA) in Grenoble, and IRD in Marseille.
[3] National Institute of Informatics, Tokyo, Japan.
[4] In two versions, in 2013 and in 2014.
[5] Centre Franco-Indien pour la Promotion de la Recherche Avancée.
[6] From Mumbai, Chennai, Bangalore, Hyderabad, and Grenoble.
[7] More than 500K tweets originate from India every day.
[8] Like for English tweets, there are many disfluencies, and a very large vocabulary, with a large and evolving set of named entities; but about 5% of Indian tweets exhibit some level of code-mixing (2-3 languages in 1 tweet).
[9] List of languages in the Eighth Schedule to the Constitution of India
[10] Its direction is at Hyderabad.
[11] Center for Indian Language Technology.
[12] Indian Institute of Technology Bombay, located in Powai, near Mumbai (earlier called Bombay).
[13] Starting from the original (English) Princeton WordNet, CFILT and its partners have translated it into Hindi and then other Indian languages, thereby adapting the WordNet synsets to the word senses in each language, and keeping trace of the translation links to obtain more precision for meaning-related tasks such as WSD and MT.

worked for several years on Statistical Machine Translation, which makes use of Machine Learning. At the same time, we noticed that it was next to impossible to get very large open source classical corpora in Indian languages, such as news, books, technical documentation, etc. By contrast, tweets represent an enormous volume, every day, and can be freely accessed using the TWITTER API.

After a process that will be detailed later in Chapter I, we chose as our main topic the construction of systems for *Helping Understand Foreign Tweets*, abbreviated as « HUFTweets ». This problem has a much larger applicability than tweet-based recommender systems, which would need access to adequate large knowledge bases. It is also less ambitious than the problem to build good enough MT systems for tweets, which, we will argue, is unsolvable, for practical as well as for theoretical reasons, irrespective of the MT technology employed.

By contrast, helping users understand foreign tweets, in the context of spontaneous « all domains » tweets, would be quite useful in practice, and our hypothesis is that it could be done by using « multiple, personalizable and interactive active reading ». In some preliminary experiments, we have seen that, when 20% of tweets are non-understandable in source, about 60% more (2/3 of the rest) become non-understandable when machine-translated. Users would doubt very much that they could find some useful information in a collection of machine-translated tweets, if 80% of them made no sense at all. What we want to show is that, using an active reading presentation of multiple word-by-word dictionary-based translations, we could reach an understandability level of, say, 60% and not 20%. To demonstrate that, and also to evaluate the minimum understandability ratio that users would find "still usable", it is necessary to build a concrete tool. For the purposes of this PhD, it has been done, at the level of a PC-based and browser-based prototype, SUFT-1, which has been used to make first experimental evaluations.

Which measures to use for our evaluations was not clear at the beginning: we wanted some task-related measures, but the classical measures used in MT evaluation could not be used: (1) no objective measure using references can be used in the absence of references, post-editing time also cannot be used, as there is no realistic setting in which MT-ed tweets would be post-edited; (2) classical subjective measures also cannot be used, because adequacy supposes understanding the source, and fluency is clearly not a good criterion for word-for-word multiple translation. We have introduced two task-related measures, one subjective and one objective, which have a negligible cost and can be used in « continuous evaluations on open data », as opposed to classical measures based on test sets related to closed learning sets. The subjective one is the *understandability ratio*[14], and the objective one is the *understandability decision time*, that is, the time it takes to decide whether a tweet is understandable or not.

In this thesis, we also demonstrate the possibility of building (and using) very large lexicons (lexical and multilingual coverage) and to compile (off-line) efficient multilingual lemmatizers, and even full-fledged morphological analyzers that are able to compute features such as gender, number, case, person, tense for better understanding in the AR mode. On the software engineering side, because our resources for development were quite limited, and although we fully acknowledge the necessity of building HUFTweets systems that would run on tablets and mobile phones, and would be able to use web-based servers as well as to work offline[15], our SUFT-1 prototype works on PCs and requires a connection to Internet for some

[14] A tweet is labeled as understandable if the user judges it makes some sense, otherwise it is non-understandable.
The ratio is the percentage of tweets labeled as understandable.
[15] using a small amount of resources (like JIBBIGO & other apps for speech MT).

functions. But that prototype can still be used to perform some experiments and evaluations, and it will be useful in the future to investigate some interesting questions such as:

Question 1: Does Active Reading really improve understandability of foreign tweets, and if so by how much?

Question 2: Is it useful to show an MT proposal alongside an Active Reading presentation?

Question 3: What can be done in a SUFT in the case of OOV words?

Question 4: If we incorporate NEs in the AR module, will it help better elicit the context of the tweet or the tweet translation?

Question 5: Will the incorporation of NEs in the AR module help get around the problem of the large vocabulary coverage inherent in the tweets?

Question 6: How to measure whether SUFT would be useful for also helping people who want to progress in their knowledge of the SL?

The remaining part of the thesis is organized as follows.

In Chapter I, we present in more detail the general scientific context and our motivations for working on helps for understanding foreign tweets. We also describe some preliminary experiments, give a detailed rationale for our two measures, present schematic illustrations of the possible types of AR interfaces, and general requirements of SUFT. Chapter II is dedicated to the presentation of the design of our prototype, called SUFT-1. In Chapter III, we describe how we developed a large and potentially (intrinsically) multilingual morphological analyzer for Hindi. Chapter IV describes the actual system in detail, evaluates its components, and presents 3 end-to-end experiments. We then conclude by summarizing our contributions and proposing some perspectives for further development, experimentation and research.

Chapter I Context and motivations for Helping Understand Foreign Tweets (HUFTweets)

Introduction

In this chapter, we present in some detail the state of the art in processing textual social media data, and in particular tweets. From a review of the literature and from some preliminary experiments we did by applying MT to Indian tweets, we conclude that current methods in MT research are inadequate for processing multilingual social media data and perform especially poorly on non-traditional free-form short texts such as tweets. The challenges arise on account of the high variability of quality in user-generated spontaneous texts and, due to the metadata forms (e.g. hashtags) and OOV tokens contained in the tweets.

Furthermore, the evaluation measures to assess MT performance on social media texts are (often at the same time) ill-conceived, unrealistic, impractical, or undeveloped. Despite the few studies on overcoming these problems, MT as a tool for understanding tweets in foreign languages still seems to be inadequate (Farzindar & Inkpen, 2015), not to speak about their dissemination, that would imply some guarantee of quality (here fidelity).

Given the recent growth of multilingual tweet exchanges around the world, it is then interesting to define another goal, that of *helping users understand tweets in foreign languages* (HUFTweets), and to show that this goal could be attained with a sufficient "quality of service", even for spontaneous, disfluent and code-mixed tweets. We also introduce two task-related measures[16], one subjective, the *understandability ratio*, and the other objective, the *understandability decision time*.

In the first section, we describe the general context coming from NLP on social media data. In the second one, we argue that MT is not a good approach for HUFT, at least for spontaneous tweets. In the third section, we give a set of requirements for a SUFT that would meet the practical constraints mentioned above, and also be usable to make experiments and evaluations, and to answer questions such as:

Question 1: Does Active Reading really improve understandability of foreign tweets, and if so by how much?

Question 2: Is it useful to show an MT proposal alongside an Active Reading presentation?

I.1 General context

We first situate our research in the context of NLP for social media data, brilliantly presented in a recent book (Farzindar & Inkpen, 2015). Then, we analyze three different situations in which UFTweets is important, with various degrees of urgency and quality. In subsection I.1.3, we mention other work on tweet-related research topics.

I.1.1 The overall domain : NLP on social media data

I.1.1.1 General considerations and lessons from *Farzindar & Inkpen*

The explosive growth in the social media domain combined with recent technological advances present several scientific challenges for NLP research. (Farzindar & Inkpen, 2015)

[16] already briefly described in the Introduction.

present a recent overall account, discussing powerful methods and algorithms for language processing applicable to free-form multilingual social media text.

They emphasise the need to adapt the techniques presently in use for semantic analysis of clean texts to such non-conventional text, and the applicability of such semantic analyses to areas such as social media analytics, health, security, disaster response management, business intelligence and entertainment.

The authors also draw attention to the rich research potential for multilingual processing of such user-generated content, and review various evaluation benchmarks used in emerging forums on language and semantic processing of social media data (SemEval, EMNLP, « Making sense of microposts » workshop series).

I.1.1.2 **Activities concerning multilingual aspects of social media**

Multilingual processing of social media data is interesting because of the challenges inherent in the text characteristics (code-switching, mixed languages). (Lui & Baldwin, 2014) look at adapting NLP techniques to short informal texts, especially TWITTER messages. Furthermore, studies on tools for language identification on less common languages (Bergsma, McNamee, Bagdouri, Fink, & Wilson, 2012) and early stage experiments on dialect identification for Arabic (Habash, 2010) lead to motivations for investigating the potential of MT of social media texts and evaluation mechanisms/measures thereof.

I.1.1.3 **Research on MT of tweets**

Research on translation of tweets, which are very short texts (less than 140 characters in principle) are few due to various reasons. We look at previous attempts and some reasons why MT is not and cannot be adequate for helping people understand foreign tweets, or at least spontaneous foreign tweets.

I.1.1.3.1 Previous attempts

Manual translation of tweets has been attempted by the BRIDGE part[17] of the MEEDAN[18] project. It must be stressed here that their goal is not to help users simply understand foreign tweets, but to help translators (who necessarily know the source language quite well) produce good translation. Here is an overview of BRIDGE (as of August 2017).

> ***Translate quickly and accurately***
>
> *Bridge enables rapid translation of social media and the addition of important cultural, social and political notes to facilitate understanding. With intelligent tools like dictionaries and glossaries, Bridge helps you translate efficiently and with confidence.*
>
> ***Activate your language community***
>
> *All content is portable and shareable, designed to be as seamless as sharing native social media. Open up new connections across linguistic, cultural and network divides.*
>
> ***Collaborate and build your skills***
>
> *The best translations involve a variety of skills — copy editing, grammar, spelling, a grasp of meaning and nuance. Bridge helps translators work collaboratively, with their individual contributions highlighted and recognized.*

[17] https://meedan.com/en/bridge/
[18] https://meedan.com/en/

History

Meedan has a vision of a more crosslingual Internet, and we have worked with hundreds of people around the world to translate millions of words into a half dozen languages. We are committed to the potential of translation as a social good, both at home and globally, with a particular focus on enabling communities to support global journalism and civic engagement.

Since 2006, Meedan has led a number of groundbreaking crowdsourced translation projects, including News.Meedan, a crowdsourced translation site for journalism across the Arabic- and English-speaking webs and Speak2Tweet, an effort to translate voice messages from the Egyptian revolution. Members of Meedan bring a rich array of professional experience in crowdsourced and digital translation, including Ai Weiwei English, a dedicated translation site for the Chinese artist-activist's Twitter.

Our current collaboration with Pulitzer Prize-winning journalist Paul Salopek's Out of Eden Walk project aims to translate tweets situated in the vicinity of his 7-year walk around the world. We are pleased to partner with the DOLLY Project and Translators Without Borders on this effort.

Our goal is quite different: help users understand on the spot tweets written in a language they know not at all, or at a very limited level. We are then looking for fully automatic helps. The first that comes to mind is of course Machine Translation.

(Gimpel et al., 2011) claimed that very few studies had focused on automatic translation, without actually mentioning any. We examined that claim and agree with the conclusion.

We found earlier studies that addressed tweet MT, commented on the tasks of collecting bilingual tweets, and developed two systems (German-English (L. E. Jehl, 2010), Arabic-English (L. Jehl, Hieber, & Riezler, 2012)).

"Microblogging services such as Twitter have become popular media for real-time user-created news reporting. Such communication often happens in parallel in different languages, e.g., microblog posts related to the same events of the Arab spring were written in Arabic and in English. The goal of this paper is to exploit this parallelism in order to eliminate the main bottleneck in automatic Twitter translation, namely the lack of bilingual sentence pairs for training SMT systems. We show that translation-based cross-lingual information retrieval can retrieve microblog messages across languages that are similar enough to be used to train a standard phrase-based SMT pipeline. Our method outperforms other approaches to domain adaptation for SMT such as language model adaptation, meta-parameter tuning, or self-translation."

In a similar attempt, (Ling, Xiang, Dyer, Black, & Trancoso, 2013) extracted 1 million Chinese-English parallel segments using re-tweeted messages, in order to build and evaluate existing MT systems on tweets.

Table 1: MT evaluation results from (Ling et al., 2013)

	Syndicate		Weibo	
	ZH-EN	EN-ZH	ZH-EN	EN-ZH
FBIS	9.4	18.6	10.4	12.3
NIST	11.5	21.2	11.4	13.9
Weibo	8.75	15.9	15.7	17.2
FBIS+Weibo	11.7	19.2	16.5	17.8
NIST+Weibo	**13.3**	**21.5**	**16.9**	**17.9**

The authors performed MT experiments on news and microblog data and reported an increase of just about 3 to 4 BLEU points as shown in Table 1 above.

So far, when evaluated by subjective measures, attempts at applying existing MT systems to streams of tweets or at building tweet-oriented MT systems have also yielded quite bad results. In any case, BLEU and other objective reference-based measures are not usable on such streams due to the lack of reference translations.

Maybe specialized MT systems could be developed, but there are practical as well as theoretical obstacles to such an endeavour.

I.1.1.3.2 Practical reasons why MT is inadequate for tweets

Applying classical MT techniques to tweet translations could be good enough for controlled or well-formed tweets, but there is no good perspective yet for adapting them to build "good enough" MT systems for spontaneous and informal tweets.

From the above analysis of the state of the art, we conclude that domain adaptation techniques are not usable in practice to develop useful SMT systems for tweets. The main reasons are the lack of large enough good quality parallel tweet corpora, and the unavoidable lack or lexical or lexico-semantic resources. Indeed, even in a given "domain" (translation context), the basic vocabulary is very large, if one counts not only simple words, but also simple or compound terms, and the even larger set of named entities (Farzindar & Inkpen, 2015).

The lack of large corpora is a fatal obstacle when one wants to build an empirical MT system (SMT, EBMT). It is not so fatal when one builds an expert MT system (LBMT), as small corpora suffice, but in that case the lack of lexical resources is an unsurmountable obstacle.

Practical limits imposed by the TWITTER platform and constraints on obtaining large amount of tweets for research constitute another hindrance to an extended research on tweets in multiple languages.

I.1.1.3.3 Theoretical reasons why MT is inadequate for tweets

For informal tweet texts, handling out of vocabulary (OOV) tokens seems to be a common obstacle in system improvement. (L. E. Jehl, 2010) addressed the task of English-German tweet translation, and remarked that proper treatment of unknown words is very important : even if the texts are very short (less than 140 characters), they touch all domains and their vocabulary is immense[19]. In addition, the input is very noisy, and often contains code-switching (Dey & Fung, 2014) as well as ungrammaticalities and disfluencies. Also, not only is the lexical coverage quite large, but there is a huge number of named entities.

[19] One of our tasks will actually be to evaluate the size of the vocabularies of various interesting subsets of tweets.

Hence, the problem of translating tweets so that they can at least be understood, even if the translations were not quite grammatical, is a very hard problem. To date, we know of no specific efforts to solve it. However, it should be solved, because there are real needs, as detailed in the next subsection.

I.1.1.3.4 Help to understand rather than try to produce good translations?

Information carried by tweets is often important to potential foreign readers in various situations, as detailed below (p. 23). But relevant tweets have to be filtered out from the enormous amount produced every day (500M per day in the world, 500K per day in India), and then translated, or at least made understandable.

Translation (manual or automatic) would theoretically be the best way for understanding foreign language tweets. However, due to the practical and theoretical limitations for tweet MT (I.1.1.3), that goal seems unattainable. That is why we attempt to address it differently.

We hypothesize that the problem of making foreign tweets understandable could be solved by « lowering the goal », that is, not by translation alone, but by combining MT with some *active reading* (AR) presentation resulting from a kind of *multiple pidgin translation* (Harris, 1976), or by AR presentation alone, in which case AR would be complemented by MT only if it would be felt useful by users for the situation at hand (subset of tweets, languages, type of sublanguage).

Here are two examples (Lao→French and Japanese→French) showing that AR can indeed be a good understanding help for a person knowing almost nothing of the language at hand.

I.1.1.3.5 Example of a Lao→French AR presentation

Figure 1: Active reading layout in laosoftware.com (for Lao-French)

As shown in Figure 1, in the AR presentation of the software created by Vincent Berment, author of many tools to process Lao on PCs, under OFFICE and on the Web, the text is automatically segmented into words, and then displayed vertically, with translations coming from the dictionary on the right.

The fact that all possibilities are shown at the same time is quite good for people having a very low proficiency level in Lao, but having some basic knowledge about grammar and word order.
An interesting feature that could be added to help "reconstruct" the meaning is to highlight the equivalents "guessed" by the user. Another idea would be to let a selected equivalent shift to the left, in the first position.

I.1.1.3.6 Example of a Japanese→French AR presentation

This other interface as shown in Figure 2 is the CESSELIN dictionary interface for French readers built by M. Mangeot in 2015. Since that time, the dictionary has expanded from 85000 to 145000 entries, and has been corrected and completed collaboratively on the web.

The user copies some Japanese text in the window at the centre and the text is then presented at the bottom of the screen, segmented into words, with the pronunciation shown above it in furigana (small kanas) or in romaji (Latin transcription). When the user puts the pointer on a word, information from the dictionary appears. The principle is to *show all possible information for one word at a time*. If a collocation (bigram, trigram, etc.) contained in the dictionary is present in the text, its information appears with that of the simple word that is the "anchor" element of the collocation.

This tool is quite useful for native speakers of French having some proficiency in Japanese. For those with no proficiency at all in Japanese, it is more difficult as they have to memorize the possible meanings of already seen words. A main point is that it has a very large coverage.

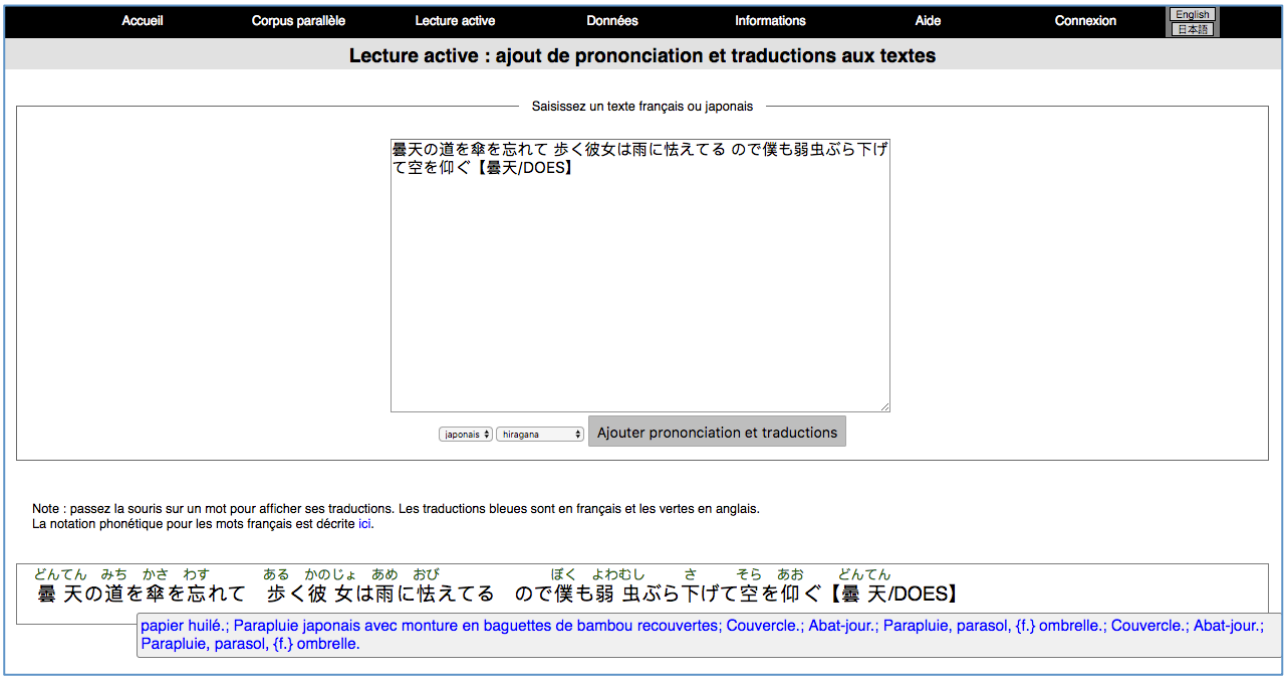

Figure 2: Active reading layout in M. Mangeot's tool (for Japanese-French)

I.1.2 Various needs for understanding foreign tweets

To assess the usefulness of such a mechanism, we first need to answer the following questions and to make more precise what we can understand by *quality* in such a context.

1. What are these needs or use cases?
2. What are the corresponding requirements in terms of quality?
3. How to measure quality?

We identify three types of needs for understanding tweets and inherent problems in each corresponding context. In due course, we propose to combine *understandability ratio* and *understandability decision latency* as a two-pronged quality measure that is both simple and task-oriented.

I.1.2.1 Daily life situations (tourists): no urgency but need for good understandability ratio

Use case: There is a recognised need of tourists (in India, in particular) to make sense of spontaneous tweets, concerning for instance, recent and opinionated content on various local events or tourist destinations. However, although there is no urgency in this situation, and no real need for very faithful translations, as quality could be compensated by quantity, there is certainly a need to get an understandability ratio felt as "not unusable". We felt that we would need the understandability ratio to be at least 50-60% in order not to drop the idea of finding useful information from the foreign tweets.

Problem 1: Spontaneous tweets exhibit a high non-understandability ratio, considerably multiplied by using off-the-shelf MT.

A study by (André, Bernstein, & Luther, 2012) shows that out of a sample of 40K tweets, 25% are rated and perceived by users as "not worth reading", because "they cannot make sense of them".

A preliminary experiment done by us in Indian→English with monolingual and mixed-code tweets gave a non-understandability ratio of 20% in source, and of 80% after MT.

Another important factor is that such user-generated texts are often disfluent, or simply un-understandable, hence difficult or impossible to understand by a native speaker of the SL.

Problem 2: Spontaneous tweets exhibit a high degree of code-mixing, agrammaticalities, typographical errors and have an immensely large coverage[20] in terms of vocabulary (named entities for instance).

The abundance of research on pre-processing of tweets corroborates the need to handle out-of-vocabulary tokens of several types: hashtags, usernames, contractions and emoticons.

[20] In our contexts, we found vocabularies of more than 300K units.

I.1.2.2 Crisis situations: specific domain, urgency, reliability is critical

Context: For foreigners, there is an urgent and critical need to understand announcements or information conveyed through regional language tweets in crisis situations. MT of tweets presently as a solution is inadequate as it lacks the reliability factor.

Use case 1 : Crisis situations for instance, tornadoes, earthquakes in Japan.

(M.-T. Nguyen, Kitamoto, & Nguyen, 2015) applied machine learning techniques on disaster related tweets in order to provide informative tweets to people, thereby helping them make quick and suitable decisions.

Use case 2 : The Arab Spring[21] revolution and efforts for Arabic-English tweet translations

During the Arab Spring in 2010, social media platforms like FACEBOOK and TWITTER emerged as effective information streams. Activists used them to organize and communicate internal local protests and the foreigners witnessing the situation used them for broadcasting.

In such situations, understanding Arabic tweets in English could be an important need. Since 2005, the Meedan[22] web service (presented in I.1.1.3.1 above) has been set up to translate Arabic tweets to English to spread information about Middle East issues (Farzindar & Inkpen, 2015). An example from the Meedan website is shown in Figure 3.

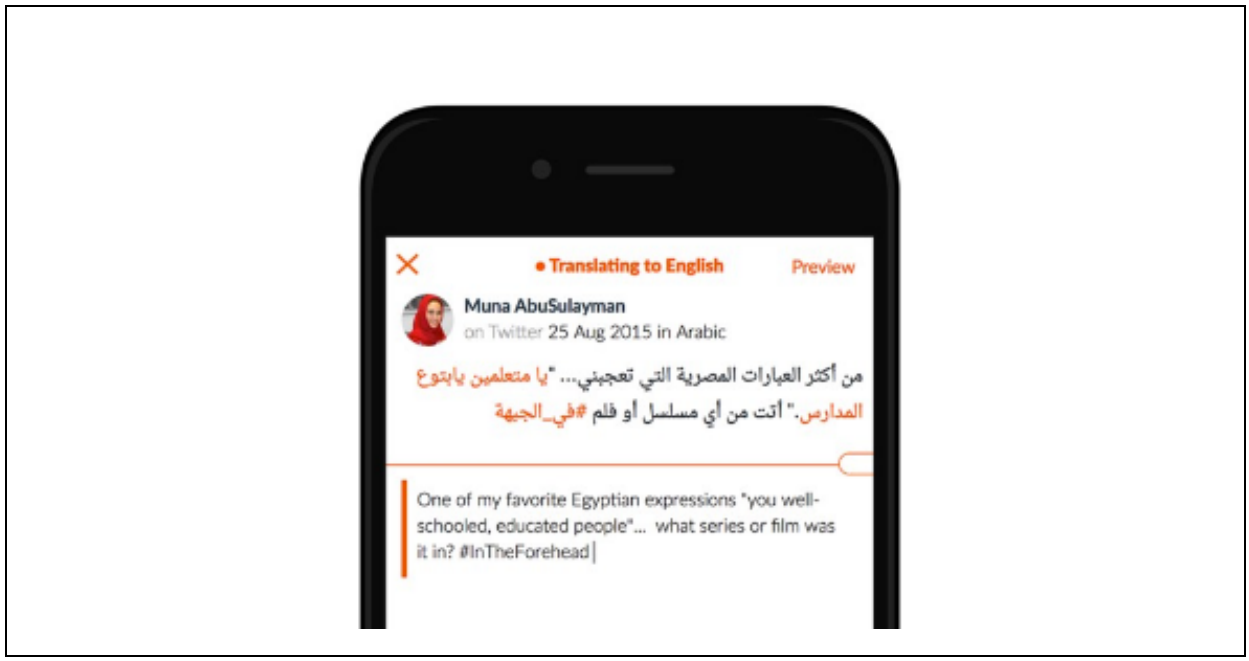

Figure 3: An Arabic tweet translated to English using Meedan translation service on a mobile phone

[21] https://en.wikipedia.org/wiki/Arab_Spring

[22] https://meedan.com/en/. Translation is performed by volunteer translators. They probably use on-line dictionaries and translation memories, but we could not find any precision on that point.

I.1.2.3 **Professional information seekers: middle urgency, high quality required**

Use case 1 : Good understanding of tweets by journalists

Tweets are a good source of opinionated content, recent trends and public viewpoints on local or international affairs. As a result, foreign journalists resort to such information streams and need to understand foreign tweets. The urgency involved and the need to understand the tweets with a certain precision is slightly less than in crisis situations.

Use case 2 : Tweets issued by the Canadian government

In Canada, where bilingualism is a legal obligation, official information must be issued both in English and French. This context implies a « semi-urgent » need where to « make sense » of a tweet is not enough. (Gotti, Langlais, & Farzindar, 2013) make an effort to automatically translate government-issued tweets that are relatively well-formed and must be communicated precisely to the public. Note however that in the precise situation, it would be better to generate the tweets in both languages from the KB or domain ontology, if any. That is done since several years for weather bulletins.

The sense of the tweets in contexts such as Case 1 above has to be validated by other tweets or maybe with a recourse to online volunteer translators or at least post-editing.

Other examples for such « middle urgency » contexts include among others, news on election campaigns and economic affairs.

I.1.3 **Other research on tweets**

I.1.3.1 **NE extraction from tweets and for tweets processing**

I.1.3.1.1 *State of the art*

NE recognition and extraction from tweet texts is an important task in social media analysis and is useful for determining the location of tweets and for entity linking and disambiguation (Farzindar & Inkpen, 2015; Ritter, Clark, & Etzioni, 2011). However, (Derczynski, Ritter, Clark, & Bontcheva, 2013) show that standard NLP methods, when applied to tweet-like noisy texts for NE processing, take a large performance drop, although domain adaptation helps.

In the context of Indian language tweets, which exhibit much code-mixing with English, (Patawar & Potey, 2016) point out there are no NE recognition systems to process regional language tweets and perform NE recognition on Marathi tweets.

Concerning Hindi and Tamil, (Devi, Veena, Kumar, & Soman, 2016) extract NEs from code-mixed tweets and are also able to efficiently classify OOV tokens. We conclude from these research results that NE processing, especially for tweets, requires innovative approaches and techniques.

I.1.3.1.2 *Our contribution*

Given the lack of a well-known gazetteer list for Indian languages, we developed elementary but effective web-crawling programs to collect Hindi and Marathi NEs (Shah, 2016) with a few examples shown in Table 2 and Table 3.

Table 2: Examples from 18821 named entities: Indian names for boys and girls

Category	Examples (Total:18821)
Boys[23] (13631)	अभय (aBaya), इन्दीवर (indIvara), कैशिक (kESika), प्रसून (prasUna), फणिपति (PaNipati), + 13625 others
Girls[24] (5190)	अग्निमुखी (agnimuKI), कुसुमा (kusuma), गीतांजलि (gItAMjali), पर्विनि (parvini), मुक्ताली (muktAlI), + 5185 others

Table 3: Examples from 49917 named entities obtained from crawling Wikipedia category[25] pages: few entries from food and location category

Language	Category	Examples (Total:49917)
Hindi[26] (37333)	Food (936)	चिकन टिक्का (cikana tikkA), परांठा (parAMThA), शाही पनीर (SAHI panIra), टंगड़ी कबाब (TaMgDZI kabAba), + 932 others
	Location (26382)	बुकामाख्या मन्दिर (bukAmAKyA maMdira), ताजमहल (tAjamaHala), बौद्धनाथ (bOddhanAtha), खजुराहो (KajurAHo), पत्तदकल (pattadakala), + 26377 others
Marathi[27] (12584)	Food (561)	आमखंड (AmraKaMDa), आवळ्याचे लोणचे (AvaLyAce loNace), उसळ (usaLa), कांदे पोहे (kAMde poHe), कांद्याचे लोणचे (kAMdyAce loNace), + 556 others
	Location (484)	त्र्यंबकेश्वर (tryaMbakeSvara), महांकाळेश्वर (maHAkALeSvara), शृंगेरी (SruMgerI), अमृतेश्वर मंदिर (amruteSvara maMdira), केळझर (keLaJara), + 479 others

1.1.3.2 **Research on recommenders built from tweets**

Recommenders providing personalised or non-personalised recommendations to TWITTER users are built with various approaches (Kywe, Lim, & Zhu, 2012). Some recommenders use information within tweets for textual or semantic processing. Two approaches are broadly used.

1. Metadata from within the graph-like structure of the social network are used for twitter analytics (Yan & Li, 2012).
2. The content-based approach uses content characteristics and features (Han, Cook, Au, & Baldwin, 2014).

(J. Chen, Nairn, Nelson, Bernstein, & Chi, 2010) recommend content from information streams which are characterised as recent, user-generated and user-interactive. Interestingly, in the multilingual context, (Neubig & Duh, 2013) report on a study on the amount of information contained in each tweet across 26 languages, and (Z. Wang & Iwaihara, 2015) propose a cross-lingual tweet recommendation system, aiming at recommending meaningful Japanese tweets for English users based on their interest.

[23] collected from http://astrology.raftaar.in/baby-name/boy
[24] collected from http://astrology.raftaar.in/baby-name/girl
[25] https://en.wikipedia.org/wiki/Help:Category
[26] collected from several categories under https://hi.wikipedia.org/wiki/
[27] collected from several categories under https://mr.wikipedia.org/wiki/

For a recommender based on multilingual multiscript tweets a design has been proposed by (Shah, Boitet, & Bhattacharyya, 2015) with the relevant NLP modules and their descriptions.

I.1.3.3 **Tweet processing for studying language usages**

I.1.3.3.1 Review of research on that topic

In terms of publications, research on tweet processing for studying language usages can be seen in the following work.

(Zielinski & Bügel, 2012) studied the problems of analyzing multilingual (Romanian, Greek and Turkish) TWITTER feeds for emergency situations. (Oostdijk, 2015) investigated syntactic constructions to develop a rule-based parser for analyzing Dutch tweets. In another direction (Gauthier, Guille, Rico, & Deseille, 2015) analyzed specific sociolinguistic features like gendered uses of British swear words on TWITTER.

In terms of research activities, for instance, the CTS (Corpus-based Translation Studies) conference invites corpus-based studies for MT research.

I.1.3.3.2 Study on Gujarati tweets in Africa

Motivated by a Call for Papers for a CTS (Corpus-Based Translation Studies)[28] conference, we undertook a study on the usage of various languages in the tweets, and began by our native tongue, Gujarati. That study has not yet been published, because that conference, announced for October 2016 in Pretoria, was postponed first to April 2017, then *sine die.*

a. Rationale

Gujarati has about 40M speakers in India, and more than 500K speakers in the African region including the Comorian Islands. A field report on Indian languages in Africa has been published by (Mesthrie, 1997), and there is ongoing research[29] on Gujarati influence in Kenya and East Africa. Also, South Africa is supporting Gujarati, as well as many languages of native or immigrated minorities.

(Probyn, 2016) published a study of TWITTER trends across Africa, notably stating that 1.86 billion tweets were generated from Africa in 2015. This triggered us to perform a study on language usage by investigating the presence of tweet streams in the Gujarati language[30] originating from Africa.

b. First improductive search

We approached the task of retrieving Gujarati tweets from Africa in two different ways.

1. We retrieved tweets with 'lang:gu' and Gujarati unigrams/bigrams as queries to the TWITTER search APIs. The relevant tweets were then collected by filtering the output based on geolocation metadata (cf. I.1.1.3.3.2.{b-c}).
2. We retrieved tweets with 'African location coordinates' as queries and then filtered the output by identifying Gujarati scripted text (cf. I.1.1.3.3.2.d).

The first procedure was divided into two parts. In the first part, we searched 5 times with the query set 'QS1' containing the language operator "lang:gu" and then submitted 5 different Gujarati scripted word forms denoted by query set 'QS2'. For more details on each query submission and its result, see **Error! Reference source not found.**.{1.a, 1.b }. In the second

[28] This research domain was first presented by Alet Kruger in 2002 .
(http:// www.ajol.info/index.php/actat/article/viewFile/5455/29593).
[29] http://timesofindia.indiatimes.com/city/ahmedabad/US-researcher-looks-for-Gujarati-influence-in-Kenya-and-East-Africa/articleshow/46816299.cms.
[30] For simplicity, we do not consider transliterated forms of Gujarati in our experiments.

part, we used 50 Gujarati bigrams denoted as query set 'QS3' for further retrieval, as explained in the next section. The tweets retrieved in both cases were then filtered based on geo-location metadata.

Given the idiosyncrasies of the TWITTER REST search APIs[31], we used QS1 and QS2 simply to get a very first estimate of retrieved tweets, denoted by 'TS1' and 'TS2' respectively. However, contrary to the statistics reported by (Probyn, 2016), the amount of tweets retrieved was much lower, as shown in Table 4.

The shaded cells of Table 4 show the vocabulary sizes of TS1, TS2 and the number of 'hashtags' and 'usernames' extracted from the vocabulary. #Filtered terms are terms from the vocabulary after removal of 'RT', URLs, hashtags and usernames. #Filtered terms are then grouped as ASCII, non-ASCII and mixed code terms.

Table 4: Vocabulary size and number of different types of terms in the vocabulary

	Unique tweets / Total tweets retrieved	Vocabulary size and number of different types of terms in the vocabulary							Code-mixing %
		#Vocabulary	#Hash tags	#Usernames	#Filtered	#ASCII	#Non-ASCII	#Mixed code	#ASCII *100/ #Filtered
(TS1) using lang:gu	9731/ 13948	33572	1017	2531	25174	1184	23431	559	4.7
(TS2) using 5 unigrams	1616/ 3399	11094	311	561	9365	496	8770	99	5.3

Figure 4 below shows a few examples from the relevant Gujarati tweets[32].

(1) કયાંક ખુશી છે.. કયાંક વ્યથા છે.. અહી તો ચહરે ચહરે એક કથા છે.. #હિરેન

(2) @sanmistryious: #ગુજરાતએટલે જ્યાં સરકારી બસો માં સરખી રીતે નંબર પ્લેટ લગાડવા કરતા પાનમસાલાની જાહેરાતો લગાડવી વધુ જરૂરી છે. #AMTS

(3) RT @pinakin_joshi: હું સિંહ જવો છુ, જે પોતાના જગંલ માં આરામ કરે છે, એકલો પણ અભય, મને આસપાસ કુદા કુદ કરતા વાંદરા ઓ થી ફરે નથી પડતો.

Figure 4: Gujarati tweets obtained from Africa (sparingly code-mixed with Roman script)

As a last step, we separated the geo-enabled tweets (per user) from TS1 and TS2 and then programmatically selected tweets with African location names. TS1 and TS2 had 1681 (12.1%) and 547 (16.1%) geo-enabled tweets respectively. TS1 contained 42 relevant tweets from 3 users belonging to Kampala, Johannesburg and Lubumbashi, while TS2 contained 71 relevant tweets from 2 users belonging to Lubumbashi and Mbale.

Interestingly, we found no Gujarati tweets coming from the Comorian islands, although the Gujarati community there is rather numerous.

[31] https://dev.twitter.com/rest/public/search.
[32] The main script used in the examples is Gujarati; https://en.wikipedia.org/wiki/Gujarati_alphabet.

c. Findings when using bigrams

Based on the findings and sparse results retrieved above, we extended the above method to submit the 50 most frequent Gujarati bigrams as query set 'QS3' from among the resources provided by (Scannell, 2007). For more details on each query submission and its result, see **Error! Reference source not found.**.1.c. As shown in Table 5, we obtained a meagre 5621 (38.8%) unique tweets after removing duplicates from 14487 retrieved tweets denoted as 'TS3'.

Table 5: Vocabulary characteristics of tweets obtained by querying 50 Gujarati bigrams

	Unique tweets / Total tweets retrieved	Vocabulary size and the number of different types of terms in the vocabulary							Code-mixing %
		#Vocabulary	#Hash tags	#Usernames	#Filtered	#ASCII	#Non-ASCII	#Mixed code	#ASCII *100/ #Filtered
(TS3) using 50 bigrams	5621/ 14487	21597	634	940	18332	925	17179	228	5.05

The number of geo-enabled tweets in TS3 was 869 (6%), but surprisingly only 4 tweets by 2 users from Maputo and Mbale were found to be relevant. Even though their public TWITTER profile indicated a total of 6508 posted tweets, the TWITTER extraction mechanism allowed us to get only 2-7% of these 6508 tweets, because of the recency constraints of the API.

From the first method of "querying and then filtering by location", we note that if we simply query through TWITTER search APIs, the retrieval rate is bad and highly variable because it depends on the amount of recent relevant tweets generated by the community.

We also see that the code-mixing ratio in the relatively small amount of Gujarati tweets retrieved by this method seems to be around 5%.

d. Method still improductive when extended to all of Africa

In the second approach, we obtained tweets by querying with location coordinates for 23 cities from 16 countries in Africa. Most of them are those with the largest population, and the others are capitals or commercial centres. For more details on each query submission and its result, see **Error! Reference source not found.**.2 We then searched for Gujarati scripted text within the tweets obtained denoted as 'TS4'.

Table 6: Vocabulary characteristics of tweets obtained by querying 23 geo-coordinates

	Unique tweets / Total tweets retrieved	Vocabulary size and number of different types of terms in the vocabulary							Code-mixing %
		#Vocabulary	#Hash tags	#User names	#Filtered	#ASCII	#Non-ASCII	#Mixed code	#ASCII *100 / #Filtered
(TS4) 23AfrLoc	68369/ 86176	184654	9987	25156	120081	100434	10187	9460	83.64

The number of geo-enabled tweets was 22538 (26.2% of the total retrieved). Unfortunately, the second method proved to be even more improductive than the first: we found no Gujarati scripted text at all! The code-mixing proportion of 83.64% as seen from Table 6 is explained by a residual mix of Japanese, Arabic and Russian scripts.

I.2 MT is not sufficient, but multiple pidgin MT might be

I.2.1 Evaluation methodology and possible settings

I.2.1.1 Rationale for the choice of evaluation measures

We have seen that classical MT is insufficient as main component of a system for helping understand foreign tweets, because it reduces the understandability ratio to a very low level, namely, in one of our preliminary experiments, from 80% to 20% in the case of Indian tweets translated by GT into English.

Based on some convincing examples, we embarked on the project to build such a system (abbreviated as SUFT) by basing it on a less ambitious kind of MT, namely "multiple pidgin MT" (a term coined by B. Harris in 1970), or, more precisely, on multiple word- and term-based MT used in a user-friendly "active reading" environment.

Which quality measures can we envisage in this context? First, we want to propose task-related measures, which can be used not only during development of the system, but during its whole operational life. Second, we would like to propose at least one subjective measure and one objective measure.

Subjective measures are those that are based on human judgments. Many have been proposed and used in MT, notably fidelity, grammaticality, terminological consistency, etc.[33]. Since the advent of empirical MT (mainly SMT), two new measures have been introduced, *adequacy* and *fluency.*

Adequacy is very badly defined as the perceived percentage of meaning transferred in the automatic output: nobody so far has been able to define a reasonable meaning quantification, and anyway the proposed scale (from 1 to 5) is inadequate, as it cannot account for *countermeanings,* which should give rise to negative scores. Moreover, in evaluation campaigns, adequacy has been „measured" by using „reference translations" in the *target language,* for the very dubious reason that judges knowing the source language would be too expensive or rare, or both. But, by doing this, distinctions that are not made in the SL[34] and have to be made in the target language are made (like number or gender in jp-en), so that a perfectly good MT output using the other possibility will get a bad score. Also, adequacy can evidently not be measured for segments that are not understandable in the source language. We could discard them from the counts, but, in the case of foreign tweets, anyway, we cannot hope to get reference translations when our SUFT system will be running, hence adequacy, even if it would be improved, cannot be used.

That is why we propose as subjective measure the *understandability ratio* in the target language, the best score being achieved when that rate is the same as that of the original tweets in the source language(s). We take it that this measure incorporates the appreciation of the global ergonomy of the system, which is bound to be different on PCs, tablets and mobile phones.

Definition 1: Understandability ratio.
The understandability ratio is the percentage of tweets that are understandable by a user of a certain profile, in a certain context, e.g. using the source tweets only, MT results, or Active Reading aid.

[33] See for example the 2 JEIDA studies (Isahara, 1995) and (Nomura & Isahara, 1992) led by late Pr. Nomura in Japan.
[34] SL: source language, TL: target language.

In MT, objective measures are based on similarity with reference translations (BLEU, NIST, METEOR, ORANGE, TER, HTER, mPER[35], or on the human time to reach some goal, like producing a good output by post-editing the MT output, or to understand up to a certain TOEIC-like level, or to perform some other task, like booking a hotel or a plane or ordering in a restaurant (Boitet, Blanchon, Seligman, & Bellynck, 2009). In our context, as reference translations will never be produced, we are left with measuring the "human effort".

We settle for the time it takes a user to decide whether the pidgin-translated tweet makes sense to her/him or not, and call it *understandability decision time.*

Definition 2: Understandability decision time.
The Understandability decision time is the average time it takes for a user of a certain profile, in a certain context (e.g. using the source tweets only, MT results, or Active Reading aid) to decide that s/he can "make sense of it" or not.

I.2.1.2 **Factors likely to influence those measures**

Our measures depend a priori on various factors, such as:

1. Initial competence of user in source language (SL).
2. Lexical coverage of the system (in SL and SL → TL).
3. Parts of the interface made accessible to the user/evaluator. (e.g. AR only, AR+MT, MT only, +/- proactive "natural dictionary".

This will guide us in choosing different user profiles for our experiments and evaluations.

I.2.1.3 **Principle of evaluation-oriented experiments**

Like (Huynh, Boitet, & Blanchon, 2008) have done for the IMAG/SECTRA system in the context of post-editing MT results in multilingual access gateways, we will design our SUFT system to incorporate evaluation in the usual operation of the system, and a component to prepare and run experiments in "real life" contexts.

Here are the steps deemed necessary to prepare evaluation-oriented experiments.

1. Collect a large enough set of tweets and display each tweet with its annotations.
2. For each tweet and in a given operational context, ask the evaluator to label each tweet as « understandable » or not, simultaneously recording the time taken for each tweet.
3. Repeat the experiment until both measures stabilize.[36]

I.2.1.4 **Possible evaluation conditions**

We first delineate the factors that determine the evaluation conditions within a SUFT and then elaborate on the various evaluation settings we propose to put as requirements for SUFT-1 in order to be able to perform real experiments with those settings.

The evaluation conditions in a SUFT are determined by:

1. use cases: "non-urgent", "very urgent", "semi-urgent needs (cf. Section I.1.2) for understanding foreign tweets.
2. user profile: role (evaluator, end-user), language competence.

[35] Mixed post-edition error rate, introduced by Christian Boitet & Mélanie Pineau in 2004, before IWSLT-04, where mPER(pe, mt) = α D_{char}(pe, mt) + (1-α) D_{word}(pe,mt) and Ω_{word}(w1, w2) = D_{char}(w1, w2) ($\Omega \in$ {i, d, x}).
[36] Not used in the experiments we performed so far, but this will be addressed in the future.

3. operational context/system ergonomy:
    a. AR environment: layouts (vertical/horizontal), user interaction (proactive dictionary), presentation elements (tooltip, dropdown).
    b. presence or absence of other MT modules: GOOGLE TRANSLATE, YANDEX
4. languages (source-target pair): "hi-en", "jp-en", "jp-fr", "mr-en" etc.
5. test sets: closed and/or open (real time or streaming).

In the list above, the *operational context* and the *user profile* directly influence our task-related measures, *languages* determine the multilingual setting and in turn the requirement of resources. The choices for *test sets* and *use cases* are made during an experimental setup.

In our context, we would like to perform evaluation experiments using SUFT-1 and so we propose the following specifications for SUFT-1, with their rationale.

Evaluators of SUFT-1 should be able to evaluate:

1. with the 3 language pairs "hi-en", "jp-en", "jp-fr".

   Reason: We have the language expertise for "hi-en" and a collaborative exchange with NII labs (Tokyo) that equipped us with "jp-{en,fr}" data resources.

2. for use cases with "non-urgent" and "very urgent" needs for understanding tweets.

   Reason: We have access to spontaneous "tourism-related" tweets for Hindi and "snow" related tweets for Japanese, and to dictionary resources we believe will yield the desirable coverage.

3. with closed and open test sets.

   Reason: Closed test sets evaluation could be possible with 'file upload' libraries and open test sets evaluation can be accommodated by implementing a search interface using TWITTER API libraries for real-time retrieval.

4. with some of the configurations constituting the operational context:
    a. horizontal layout (word-by-word).
    b. use of tooltip and dropdown for annotation presentation.
    c. possibility to select annotation.
    d. displaying combinations namely AR only, MT only and AR+MT both.

   Reason: The UIKIT web technology can help build interfaces with above configurations, especially the ACCORDION[37] element from UIKIT is best suited for the last one.

5. as evaluators.

   Reason: Mechanisms for recording and logging understandability decision and understandability decision time can be built using client/server solutions and evaluators with the language expertise can be invited to participate in the experiments.

I.2.2 **Preliminary experiments**

We have performed some preliminary experiments in several contexts involving different language pairs, to gain a better perspective on what can be obtained by using classical machine translation on tweets. We made use of the opportunity to perform them on (1) accessing Indian tweets (exhibiting some degree of code-switching) in English, and on accessing Japanese tweets (2) in English and in (3) French.

[37] https://getuikit.com/v2/docs/accordion.html

1.2.2.1 **In the « Indian-English context »**

In 2015, we performed an experiment on Hindi tweets translated to English (Shah & Boitet, 2015).

As justified above (p. 23), we estimated the quality of machine-translated tweets in terms of their *understandability* by users having no knowledge of the source language, or only very basic notions of it.

We made in fact 2 successive experiments, each with 100 Hindi tweets (with about 5% code-mixing involving English and emojis), at 1 week interval. We first evaluated ourself their understandability ratio (in their original form) and found it to be about 80% for both sets. We then had them translated into English using GOOGLE TRANSLATE (GT), and had their understandability ratio (in English) evaluated by an English speaker having no knowledge of Hindi. Understandability dropped to 20%, again for both sets.

For the sake of showing the original meaning in English and appreciating the divergences in the MT ouputs, we also post-edited the MT outputs (the understandable 80%, of course). Here, the post-editing time was considerably shorter for the second set (about 13 minutes per standard page[38]) than for the first one (about 21 mn/p). The most probable reason is that we learned to use the post-editing tool while working on the first set.

Using the quality formula from (Boitet et al., 2009) below, we arrived at scores of 56% and 73%, respectively[39]. That gives an idea of the (impossibly large) human effort that would be needed to build an empirical MT system for tweets.

$$Q = 1 - 2/100 \times \frac{T_{pe_{total-mn}}}{T_{hum_{estim-mn}}} \times T_{hum_{std\_page}}$$

The subjective assessment of the non-Indian reader was that he would not use a tweet-understanding help if the (global) understandability ratio were lower than 2/3 (66%). Considering that MT cannot translate ununderstandable tweets into understandable ones, and taking into account the experiment above, a simple computation shows that the goal of 66% understandable tweets could be reached with MT only if the MT system produced at least 80% understandable tweets on originally understandable tweets.

As it is, GOOGLE TRANSLATE produces only 1/3 (33%) understandable tweets on originally understandable tweets. The prospects of improving the understandibility ratio from 33% to 80% seem bleak, to say the least.

Here are 4 examples, all fully understandable in source, annotated with the understandability of their machine translations. A few examples of tweets not understandable in source (hi) are given in Appendix 6.

[38] A standard page is 1400 characters or 250 words long.
[39] In the French system of grades, 11.2/20 is “pass” and 14.6% is “good”.

RT @yogkr: @Gr8roma पंजाब देश का ऐसा पहला राज्य बन गया है जिसने अपनी सभी 13040 ग्राम पंचायतों को ऑनलाइन कर दिया है
GT: RTyogkr: @ Gr8roma Punjab has become the first state in the country which has good online all 13040 gram panchayats. *[Understandable]*

कुछ चमन नेता भारत को दूसरे देशों से नीचा दिखाते हैं. उसको यह नहीं पता कि तू जो कांड यहां करता है उसकी सजा उन देशों मे मौत है
GT: Some champion leaders humiliate India from other countries. He does not know that the punishment you take here is death in those countries. *[Understandable]*

RT @hindiplz: बच्चे दो तरह के होते हैं...पहले कूलर चला के रोबोट वाली आवाजें निकालते हैं
GT: RThindiplz: There are two ways ... first run cooler kids voiced retrieve the robot. *[Not understandable]*

RT @DrKumarVishwas: दूसरों की झूठी खबर को चौबीसों घंटे रगड़ कर चलाने वाले अपने बारे में आई इस सच्ची ख़बर को छू भी नहीं रहे
GT: RT @DrKumarVishwas: others misinformation about running round the clock by rubbing the touch was not the true story. *[Not understandable]*

Figure 5: Examples of evaluated tweet translations (hi-en) using Google Translate. Evaluations were done during experiments in July 2015 for the SEPLN conference.

Table 7: Change in understandability from source (hi) to GT output (en)

Understandability in source (Hindi)	Understandability from GT output	
Hindi native speaker	Evaluator (bilingual)	English speaker (not knowing source)
80%	27%	20%

I.2.2.2 **In the « Japanese-English context »**

We used a dataset of 3.2M[40] Japanese tweets kindly provided by Prof. A. Kitamoto of NII. This dataset contains a mix of weather-related tweets including emojis (emoticons). He collected them by providing the keyword '雪' (snow) as the query to the TWITTER streaming API, over a period of 1 month (February 2014).

The evaluation task data was prepared by sampling 500 Japanese tweets from the above dataset. These tweets were first evaluated at source by one native Japanese native as 90% understandable. These were then translated to English using GT for further evaluation. The evaluators were 2 English speakers (not knowing Japanese) and one Japanese native bilingual speaker.

Also, the evaluation task was the same as for (hi-en): to label the translated tweet as 'understandable' if the translation makes sense (to the evaluator). Translations which seemed partly understandable and made even the slightest sense to the evaluator were admitted as 'understandable'. We did so in order to obtain a worst case estimate of the non-understandability of translated tweets.

Despite judging the translation performance quite leniently, 25% of the MT translations were labeled as understandable by the English speaking evaluators (not knowing Japanese), and 29% understandable by the bilingual Japanese-English speaker.

Here are 4 examples of originally understandable Japanese tweets, with their GT translations and the human understandability decision.

[40]The size of the 3.2M tweet dataset is 10GB; 3KB metadata (in JSON) per tweet.

雪の備え：スマホやモバブーを満充電にしておく GT: Prepare for snow: Keep smartphones and moboboues fully charged *[Understandable]*
今日の雪明日の朝が怖いなー 多分地面凍るんやろなー_(┐「ε:」_ GT: I am scared of today's snow tomorrow morning probably freezing the ground __ (「" ε:) _ *[Understandable]*
そしてこの雪の中買い物にいきネイルをしてくる 給料日バンザイ GT: And in this snow I go to shopping and pay nail Payday Banzai *[Not understandable]*
@_yukine_fake 行ってらっしゃい雪音。気をつけてね？ GT: @ _ Yukine_fake Go and say helmet Yuki. be careful? *[Not understandable]*

Figure 6: Evaluation of tweet translation understandability (jp-en) performed by English speakers not knowing written Japanese

Table 8: Change in understandability from source (jp) to GT output (en)

Understandability in source (Japanese)	Understandability from GT output	
Japanese native speaker	Evaluator (bilingual)	English speaker (not knowing source)
90%	29%	25%

1.2.2.3 **In the « Japanese-French context »**

We used the same set of 500 Japanese tweets as above and made similar evaluations for the tweets translated from Japanese to French. Our evaluators were 3 French speakers who were not able to read Japanese. On average, they rated 31% of translations as understandable. We could not get a bilingual evaluator for this context.

雪まだ降るのかな…（；；） GT: Je me demande si la neige encore tomber ... (;;) *[Understandable]*
昼まで寝込んで起きたら寒すぎて…外が雪で…でも外出してやる！ GT: Une fois endormi en place jusqu'à midi trop froid ... il va à l'extérieur dans la neige ... mais pour sortir! *[Understandable]*
雪ですね。朝からあちこち動き回っていますが、場所によって降り方がバラバラだと実感しました。これから西に向かいますが、そこはどうなっているでしょうか安全運転行きましょう！ http://t.co/oRgPXvgYpG GT: Il neige. Je me suis déplacé autour et autour de la matinée, mais j'ai senti que la façon dont ça se passe dépend de l'endroit. Je vais me diriger vers l'ouest à partir de maintenant, mais qu'est-ce qui se passe là-bas? Conduisons prudemment! http://t.co/oRgPXvgYpG *[Not understandable]*
ばかやろー、チャリに雪とかきいてない GT: Bakayaro, pas entendu vélo de neige Toka *[Not understandable]*

Figure 7: Evaluation of tweet translation understandability (jp-en) performed by French speakers not knowing written Japanese

Note that the first tweet GT translation is not grammatical, but at the same time quite understandable… and quite wrong (it means something like "je me demande si je vais descendre avant la neige" — "I wonder whether I'll get off before the snow" and not "I wonder whether the snow will fall again").

Table 9: Change in understandability from source (jp) to GT output (fr)

Understandability in source (Japanese)	Understandability from GT output	
Japanese native speaker	Evaluator (bilingual)	French speaker (not knowing source)
90%	-	31%

I.2.3 Analysis and hypothesis

I.2.3.1 Analysis

Table 10: Changes in understandability from source to GT output (hi-en, jp-en, jp-fr)

Understandability in source	Understandability from GT output	
Native speaker	Evaluator (bilingual)	Target language speaker (not knowing source)
80% (hi)	27% (en)	20% (en)
90% (jp)	29% (en)	25% (en)
90% (jp)	—	31% (fr)

I.2.3.2 Hypothesis

Our hypothesis has 2 parts.

(1) Active reading presentations may raise the understandability ratio to the "usefulness level" of 60% (in total), if the underlying dictionaries are large enough and if morphological processing is good enough.

(2) Considering the profiles of the likely users, an interface showing all possible equivalents of all words of a tweet at the same time should give better results than an interface showing these equivalents word by word only (e.g. using a drop-down list or a tooltip).

I.2.3.3 Illustration

We present three mockups to illustrate the comparison between three interfaces. The interface and layout plays a direct role in either increasing or decreasing the understandability decision latency. The tweets put forth are SL understandable but not understandable with MT (GOOGLE TRANSLATE, here) and are all in the 'hi-en' setting.

I.2.3.3.1 Mockup with interface like V. Berment's multiple vertical interface

Source tweet : RT @Anurodh_80: मुझे चाहिए कोई बिल्कुल मेरे ही जैसा, किसी बेहतर से मेरी बनती ही नहीं

Google translation : RT @ Anurodh_80: I do not want anyone to be like me, better than me.

Human translation: I want someone just like me only, cannot get along with anyone better.

Table 11: Example 1 of vertical layout annotations for Hindi tweets

Word index	**Source tweet word forms** *(Transcription : IAST scheme)*	**AR view**
1	RT	RT
2	@Anurodh_80:	@Anurodh_80:
3	मुझे (mujhe)	to me / I (dative case)
4	चाहिए (cāhie)	want / wish / desire
5	कोई (koī)	somebody / someone
6	बिल्कुल (bilkula)	just (icl>quite), absolutely (icl>how)
7	मेरे (mere)	I (genitive) / my / me
8	ही (hī)	only (emphasis)
9	जैसा, (jaisā)	like / similar
10	किसी (kisī)	anyone / `certain(icl > anybody)` / `one(icl > human,icl > state)`
11	बेहतर (behatara)	better
12	से (se)	from / with / by / through / than
13	मेरी (merī)	mine / I (genitive) / Mary / me
14	बनती (banatī)	to become / to get along
15	ही (hī)	only (emphasis)
16	नहीं (nahīṃ)	no / not / negation
15+16	ही नहीं (hī nahīṃ)	not at all

Source tweet: RT @KapilMishraAAP: ये झूठ कहा था क्या आपने ??? https://t.co/0q0s6rRdYB

Google translation: RT @KapilMishraAAP: Was this lie told you ??? Https://t.co/0q0s6rRdYB

Human translation: Did you say this lie?

Table 12: Example 2 of vertical layout annotations for Hindi tweets

Word index	Source tweet word forms *(Transcription : IAST scheme)*	AR view
1	RT	RT
2	@KapilMishraAAP :	@KapilMishraAAP :
3	ये (ye)	these / this
4	झूठ (jhūṭha)	lie / falsehood / falsity / leasing / mendacious / `fraud(icl>thing) / falsehood(icl>lie)`
5	कहा (kahā)	said / to say (past tense)
6	था (thā)	was / to be (past tense)
7	क्या (kyā)	`what(icl>interjection) /` `what(icl>interrogative) /` `which(icl>thing)`
8	आपने (āpane)	you (pronoun) + (nominative suffix, 2s, 2p)
9	???	???
10	https://t.co/0q0s6rRdYB	https://t.co/0q0s6rRdYB

I.2.3.3.2 Mockup with an interface like M.Mangeot horizontal interface

Figure 8: View of the result of a Japanese tweet with furigana and French annotations

I.2.3.3.3 SUFT-1 multiple horizontal interface

Figure 9: Main interface of the system displaying a tweet annotation

I.3 Requirements for an AR+MT_based system for HUFTweets

I.3.1 Goals Practical side

I.3.1.1.1 Build a really usable system

The SUFT-1 system should be really usable, even though not yet on a tablet, but only on a small PC.

Although we plan for the possibility of offline operation later, we limit the prototype to an online operation.

I.3.1.1.2 Design it so that it is buildable by 1 person in the context of a PhD

The practical limit for this development is about 300 hours, hence, keeping the existing AR web applications as references[41], we plan to design and implement the system to allow integration of multiple dictionaries for a multilingual setting and keep it modular for future scalability.

Also, a case in point are the morphological analyzers for Indian languages. As the performance of the system hinges on robust lemma-based lookups and underlying dictionaries, we plan to construct a resource of large word forms from Hindi and Marathi tweets to compile efficient multilingual lemmatizers.

[41] http://jibiki.fr/reading/?

I.3.1.1.3 Include facilities for allowing for some real experimentation

Considering the above, we plan to keep the user/evaluator interface simple to allow quick experimentations with a single navigable tweet view at the top (with controls), 2 buttons for rating understandability ratio of the tweet (Yes/No), mechanism to calculate the decision time, and a text area to show the annotations in-between.

I.3.1.1.4 Make it extendable to handling some side questions

At some point we would perhaps like to study the following question:

Question 6: How to measure whether SUFT would be useful for also helping people who want to progress in their knowledge of the SL?

I.3.1.2 Research side

On the research side, we want to be able to conduct evaluation experiments on multilingual tweets using the SUFT-1 system, the results of which (in terms of understandability ratio and understandability decision time) will help estimate the usefulness and the potential of the system.

With SUFT-1, we plan to make evaluation experiments with:

1. evaluators having various linguistic profiles.
2. three multilingual settings: 'hi-en', 'jp-en' and 'jp-fr'.
3. three interface configurations, namely MT only, AR only and MT+AR.

Therefore, SUFT-1 should support management of various user profiles, mechanisms for displaying various interface configurations, and capabilities for integrating multilingual dictionaries/resources. Given the practical constraints, we decided to keep the user-management capabilities for later versions of SUFT.

Last but not least, SUFT-1 should allow experiments with open and closed test sets. This requires capabilities

1. for real time search in TWITTER and
2. for data import from prepared files.

I.3.1.3 Desirable features learnt from past & current related research on reading helps

In designing a SUFT system, we draw from past research (Harris & Hofmann, 1970) which discusses the feasibility of a principled TL translation ("pidgin" or "multiple pidgin", for brevity) when the only goal is the access to practical information. The claim is that even a SL-ignorant reader can benefit from the pidgin (word-for-word translation) approach as long as all the semantic and grammatical information in the original language is preserved. The "projection" of certain SL features in the TL helps the user gradually learn more correspondences between the pidgin words and the source language vocabulary, thereby improving readability and understandability in the long run.

The idea of using proactive word or phrase dictionaries for annotating tweets as a study guide or as a reading aid for understandability has been demonstrated to be quite efficient (e.g., M. Lafourcade's FICUS (Lafourcade & Chauché, 1998) dynamic dictionary access system for the Macintosh in 1995, or the Alexandria tool by Dominique Dutoit[42] for web pages around 2005).

The users of a SUFT could be strictly ignorant of the source language, or have a good knowledge of the subject matter and have some source language expertise (e.g. minimal

[42] http://www.tv5monde.com/TV5Site/alexandria/entretien.php

knowledge about morphology and word order). In both cases, the idea of reducing understandability efforts over a period of time by using pidgin-like annotations or an intermediate "transcoding" seems encouraging.

I.3.1.4 First approach to the design of a SUFTweet

In outlining the design of a SUFTweet we keep in mind the precise use-case, which is simply to help the user understand the tweets with the help of reading annotation. Considering the ease of use and accessibility we propose to make the system accessible through a browser (client-server architecture). Consequently, we need to ensure handling of the offline/online mode of operations.

Online mode:

1. The user can access recent tweets on-demand by querying.
2. Online dictionaries can be consulted for adding tweet annotations.

Offline mode:

1. The user can access tweets stored locally on files.
2. Local dictionaries or prepared/pre-transformed dictionaries can be consulted to produce tweet annotations.

On the other hand, a simplified SUFT interface for user and evaluator demands minimally intrusive evaluation widgets in the system. The preparation of resources will be done on a web server and the system may use the database to store evaluation data and logs.

I.3.2 General architecture

I.3.2.1 Overview

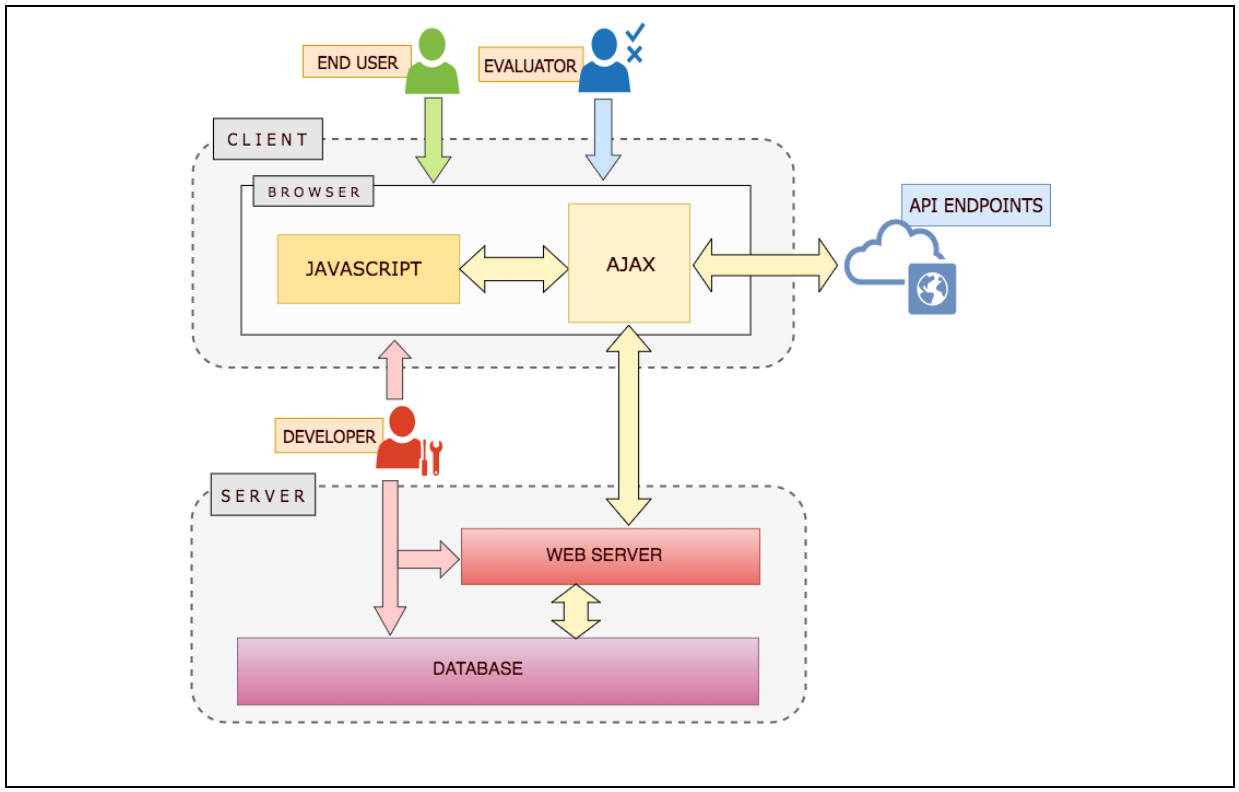

Figure 10: General functional architecture of SUFT

I.3.2.2 SUFT-1 User interface & functionalities

I.3.2.2.1 Input section: File upload + Search box The input section of the interface contain the `File upload` and `Search` functionalities to import tweets. The tweets obtained can be navigated by the `Previous` (Left arrow symbol) and `Next` (Right arrow symbol) controls.

I.3.2.2.2 Main part: text + annotations

The main part of the user/evaluator interface contains the annotations for the tweet possibly in different layouts including vertical with dropdowns and horizontal with table-like display. The section can also contain collapsible sections of MT output from different engines.

I.3.2.2.3 Controls on top (basic/advanced)

The navigation bar with controls on the top allows advanced controls to change for instance, preferences or default settings, and to switch to the evaluation interface.

I.3.2.2.4 Yes/no user feedback (for evaluation)

The bottom of the screen contains control elements (Clear/unclear buttons) to allow the user to select whether s/he has understood the tweet or not.

I.3.2.3 Evaluation module

I.3.2.3.1 Nature of evaluation

The evaluation is task-related and is characterised by two measures, understandability ratio and understandability decision time.

I.3.2.3.2 Log production

The evaluation logs must contain information about understandability (yes/no) per tweet and also the time required for the user to make his decision (understandability decision time).

I.3.2.3.3 Analysis module (on server only)

The analysis module must be deployed only on the server and must be able to provide several statistics based on processing of evaluation logs.

I.3.2.4 Data preparation module

In online mode, data is accessed by means of APIs provided, however a priori preparation of data is required, because SUFT must also function offline.

1. Offline access to local dictionaries requires data in the form of zip files to be loaded in memory.
2. Dictionaries need to be transformed into compact representations for offline use.
3. We plan to generate, in special dictionaries, articles indexed by strings that could be the result of a typing error. For example, given article `<wordform> == <content>`, we might create all articles of the form `<wordform'> == <content>` where `<wordform'>` is at edit-distance 1 of `<wordform>`.
4. Morphological analyzers need to be installed locally to help dictionary lookup through lemma-based search.

I.3.2.4.1 Dictionaries / Multilingual lexical database

Multilingual dictionaries need to be available to SUFT in an offline and online scenario[43]. In an online situation, a SUFT user could use dictionary APIs provided for instance by Jibiki[44]. In offline situations, the user might access local dictionaries transformed in a memory-loadable form, for quick and efficient access.

I.3.2.4.2 Morphological analysis data

Due to the large number of OOV[45] word forms in tweets, a lemma-based search could help SUFT increase the vocabulary coverage. To put rich morphological information in the AR tweet annotations could also be useful in certain cases. This requires the use of a full-fledged MA for the required languages. The forms along with their morphological information can be employed to good use.

I.3.2.4.3 Executable modules

For offline use, the user will need to download certain modules like MeCab[46] (word segmenter and morphological analyzer for Japanese texts), ATEF (MA for Hindi and others) and bilingual dictionaries for relevant language pairs. All of them require a PC for now, but a few could be put in a downloadable form for mobile devices.

I.3.3 Users and scenarios

I.3.3.1 Type of users

We distinguish the end users, the evaluators, and the developers.

I.3.3.1.1 End users

Foreign users including tourists and professionals seeking information are the end users of SUFT. They fall broadly under two kinds of profile: strictly target language monolingual users, and users with differing degrees of bilingual expertise (eg. knowledge of basic word order or morphology of the source language).

I.3.3.1.2 Evaluators

The evaluators of SUFT help evaluate the tweet understandability and they should have profiles similar to those of the end users. The evaluators assess the system in terms of its help to overcome or reduce non-understandability in tweets and assess the improvement in the coverage of the vocabulary.

I.3.3.1.3 Developers

The developers of SUFT debug the software and manage the lingware part, including management of dictionaries and their access mechanisms.

I.3.3.2 Scenarios

I.3.3.2.1 End User Scenario

The end-user should be able to:

1. Upload a file to import tweets.
2. Submit a "search query" to TWITTER to import tweets.

[43] We say "multilingual" because 3 lexical spaces usually come into play, those of SL, TL, and UNL.
[44] http://jibiki.fr/
[45] Out Of Vocabulary forms, that is, forms not recognized by the morphological segmenter and analyzer used.
[46] http://taku910.github.io/mecab/

3. View each tweet with annotations.
4. Navigate the tweets one at a time (previous, next).
5. Provide labels for each tweet.
6. Interact with the tweet annotations.
7. Set preferences or change default settings.
8. View machine translation of tweets.

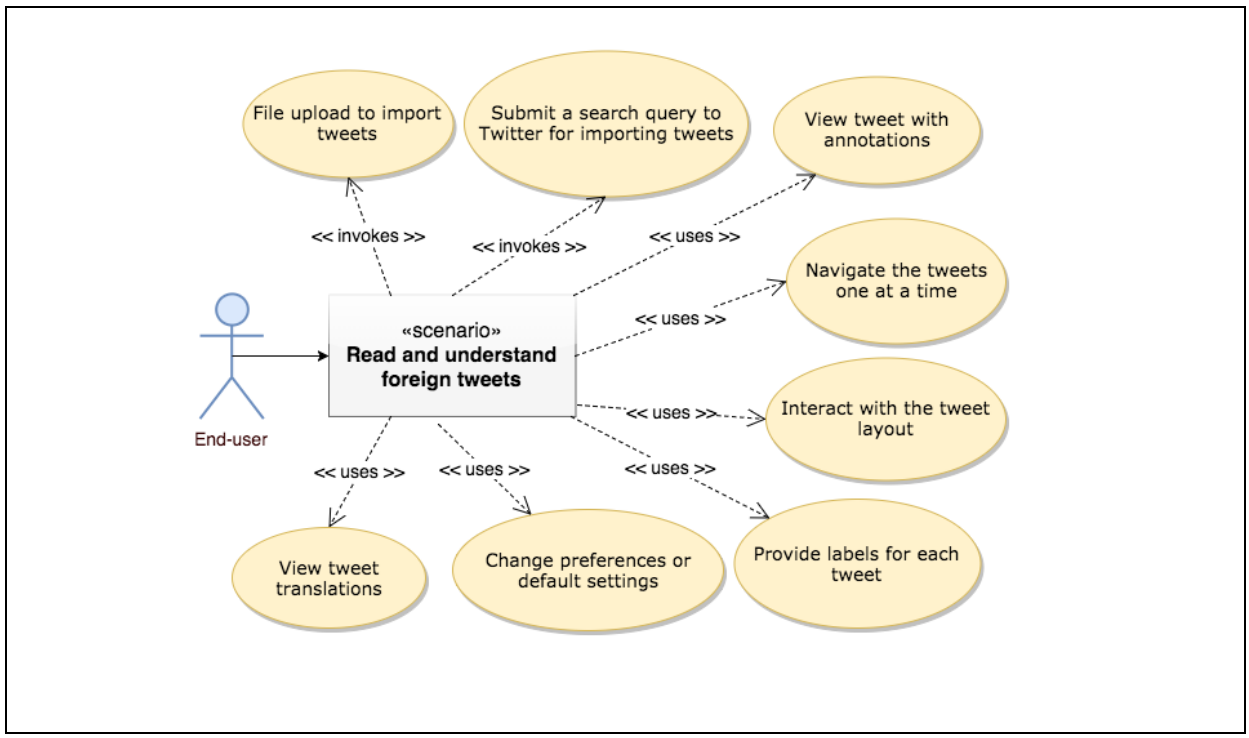

Figure 11: End-user scenario

I.3.3.2.2 Evaluator Scenario

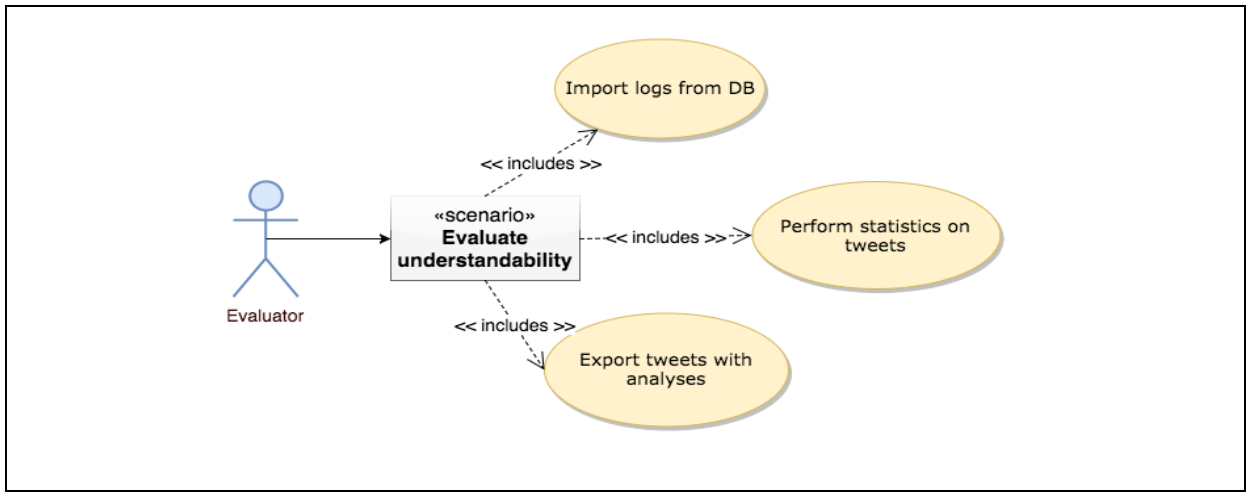

Figure 12: Evaluator scenario

The evaluator should be able to:

1. Access the logs from the database (import functionality).
2. Perform basic statistics or calculations for a set of tweets.
3. Export selected tweets with a set of analyses.

I.3.3.2.3 Developer Scenario

The developer should be able to:

1. Manage the lingware and integrate dictionaries for various languages.
2. Program and debug the SUFT software and manage related programming resources.

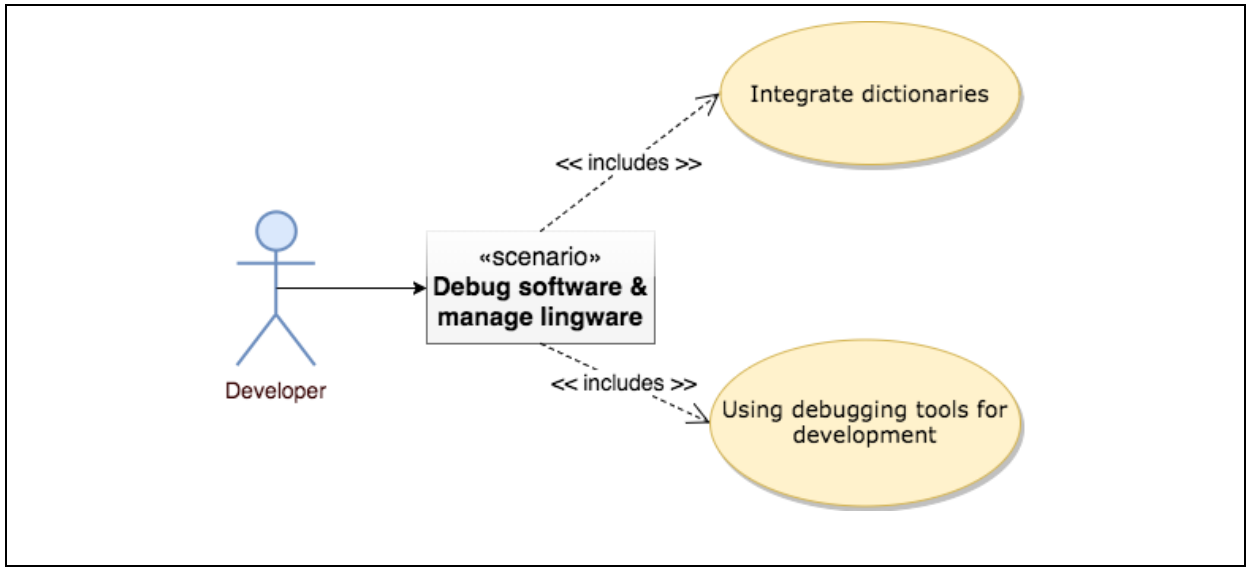

Figure 13: Developer scenario

I.3.3.3 Further requirements

I.3.3.3.1 Provide mechanisms to access TWITTER API

In order to access tweets, the system should provide mechanisms to access the TWITTER API. We identify three clear needs for various uses of tweets by different users of SUFT.

For an end-user: *a priori* select an interesting subset of tweets for information gathering or decision helping.

For an evaluator: build some collections of tweets on which to experiment, possibly with several persons (crowd-sourcing solutions like Mechanical Turks might be envisaged).

For the developer: gather large collections for computing and updating the list of word forms, or to look for new NEs within the tweets.

I.3.3.3.2 Provide a log mechanism to enable offline experiments

The evaluation of the system and its ability to help users largely depends on the interaction of the user with the system. This requires that we follow user actions and model them as logs. To enable offline experiments and evaluations a mechanism to gather logs should be provided. This must be supplemented with functions to import them in the evaluator and developer environments.

I.3.3.3.3 Provide separate interfaces for evaluator and developer

The evaluator and developer perform two separate roles and have different competence profiles. This requires that two separate interfaces containing appropriate functions and mechanisms are provided for each.

During the initial development of the system, the author undertakes both roles.

I.3.3.3.4 Provide possibilities for choosing different configurations of the general layout

The system allows several configurations for comparative experiments. For instance, the user could benefit from the presence or absence of an MT module for understanding tweets. This again demands a panel for enough controls to select the configurations.

I.3.3.3.5 Build SUFT so that ultimately the user part runs on mobile devices

It is important that the system becomes available on mobile devices as well as on PCs and tablettes. Hence, the system should be built with technologies that ensure portability across devices.

I.3.4 Implementation and performance constraints

I.3.4.1 Implementation constraints

A SUFT should be made accessible through browsers in a client-server architecture model. We propose to make use of client-side and server-side technologies (like PHP, JS, AJAX, MYSQL) alongwith the constraints they impose. Further adaptation to run the system across portable devices will require us to make use of mobile application development frameworks like Apache CORDOVA.

A SUFT should be able to access morphological analyzers and bilingual dictionaries remotely or locally, hence its implementation should ensure robust offline/online operations, including local logging of experimental data.

Also, as the user interaction is of prime importance, the implementation should provide several UI layouts for annotations, multiple experimental settings, and good user control (on font size, interface language, controls…).

I.3.4.2 Performance constraints

As far as size is concerned, SUFT should be able to handle lexical resources of at least 1M entries (knowing that Jibiki/Papillon[47] can support 2M entries and that large MA systems like ATLAS-II have a dictionary of 7M entries).
As far as time is concerned, our requirement is that:

1. processing and displaying of a tweet is almost instantaneous (less than 0.1 second).
2. searching time for tweets from SUFT should be very fast (less than 1 second).

I.3.5 Agenda, test sets, evaluations

I.3.5.1 Agenda for the first version

We planned to complete the implementation of SUFT-1 by May 2017 and then to make an evaluation with closed test sets on three language pairs, namely « hi-en », « jp-en » and « jp-fr » in the period of June-July 2017.

We followed that agenda and prepared the test sets with « hi » and « jp » tweets in order to perform evaluations on 100 tweets each for the three pairs. We planned to get the cooperation of students (internship students at NII labs, Tokyo) to evaluate the language pairs based on their expertise.

[47] http://jibiki.fr

I.3.5.2 Test sets

Closed test sets are as follows. We mention the TL alongside the SL because, at least for the small 100 tweets tests, we have prepared manual translations (by post-editing MT results) in order to make the original meaning understandable when we analyze the results.

Table 13: Test sets of tweets for Hindi, Marathi and Japanese (hi~Hindi, mr~Marathi, jp~Japanese)

hi-en	jp-en	jp-fr
hi-en-misc-100	jp-en-snow-100	jp-fr-snow-100
hi-en-misc-250K, mr-en-misc-100K	jp-en-snow-3.2M	jp-fr-snow-3.2M

We will draw the open sets by random selection from the very large collections[48] shown in Table 13 above, and, in the case of live experiments, from TWITTER streams, by making use of search queries containing advanced language operators: `{"lang:hi", "lang:mr",…}` for Hindi, Marathi (and other Indian languages) and `{"lang:jp"}` for Japanese.

I.3.5.3 Evaluations

We plan to do experiments on test sets drawn from Table 13 for "hi-en", "jp-fr" and "jp-en" depending on the participants available for evaluation.

Synthesis

After a detailed review of the literature on tweets processing, and some preliminary experiments, we outlined a method for accessing tweets in foreign languages, while not trying to produce good translations of each tweet. We proposed a method for helping users access the meaning of each tweet, in a way guessing its possible meaning, using an active reading interface, showing the possible word-for-word translation of each (simple or compound) word of the original tweet.

Although we are confident in our hypothesis, namely that such a presentation would considerably increase the understandability ratio by users having some very limited proficiency in the SL, or even none, we should now try to prove it. We have thus proposed the general requirements for a « SUFT » (System for helping Understand Foreign Tweets).

The next step is to specify and implement a SUFT that can support our claims and some experiments to answer questions about performance & design features as well as linguistic and ergonomic issues.

[48] The 'jp-en-snow-100' and the 'jp-fr-snow-100' sets contain the same sample of Japanese tweets.

Chapter II Design of SUFT-1

Introduction

In this chapter, we present the detailed specification for SUFT-1, the first version of a generic SUFT (System for helping Understand Foreign Tweets), to be used as an experimentation and evaluation platform.

We have tried to follow the best practices of software engineering, because starting from good specifications is crucial to produce an implementation that, even if it is still a prototype, it has the ergonomy and performance envisaged for the final system. Indeed, these two factors will directly influence the understandability ratio and the understandability decision time.

We then develop the external and internal specifications for SUFT-1, the prototype implemented for this thesis, keeping in mind the various use cases and communication exchanges of the various modules with the external third party APIs or the inter-component interactions.

We define APIs for all specific objectives. In particular, we include here the specifications and APIs of the tweet-related programs we have developed for extracting and filtering multilingual tweets. Last but not least, we have specified all modules by paying special attention to their efficiency, especially for the presentation and manipulations of the AR annotations as interactive confusion graphs, and for dictionary lookup.

In Section 1, we develop the external specification of the user interface, the evaluation module, the dictionary management module and the controller module. Section 2 is dedicated to the internal specifications, presented in the same order, with some additions. The third section presents the interesting aspects of our implementation, including mechanisms for efficient memory processing, annotation graph presentation and manipulation, as well as features of our programs for extracting and filtering tweets.

II.1 External specifications

We specify here the essential parts of SUFT-1: the main module, the evaluation module, the dictionary module, and the controller module.

II.1.1 SUFT-1 main User Interface

In the following subsections, we present the visual interface elements, their semantics (functional aspects), and the APIs.

II.1.1.1 Visual Interface

II.1.1.1.1 Screen display

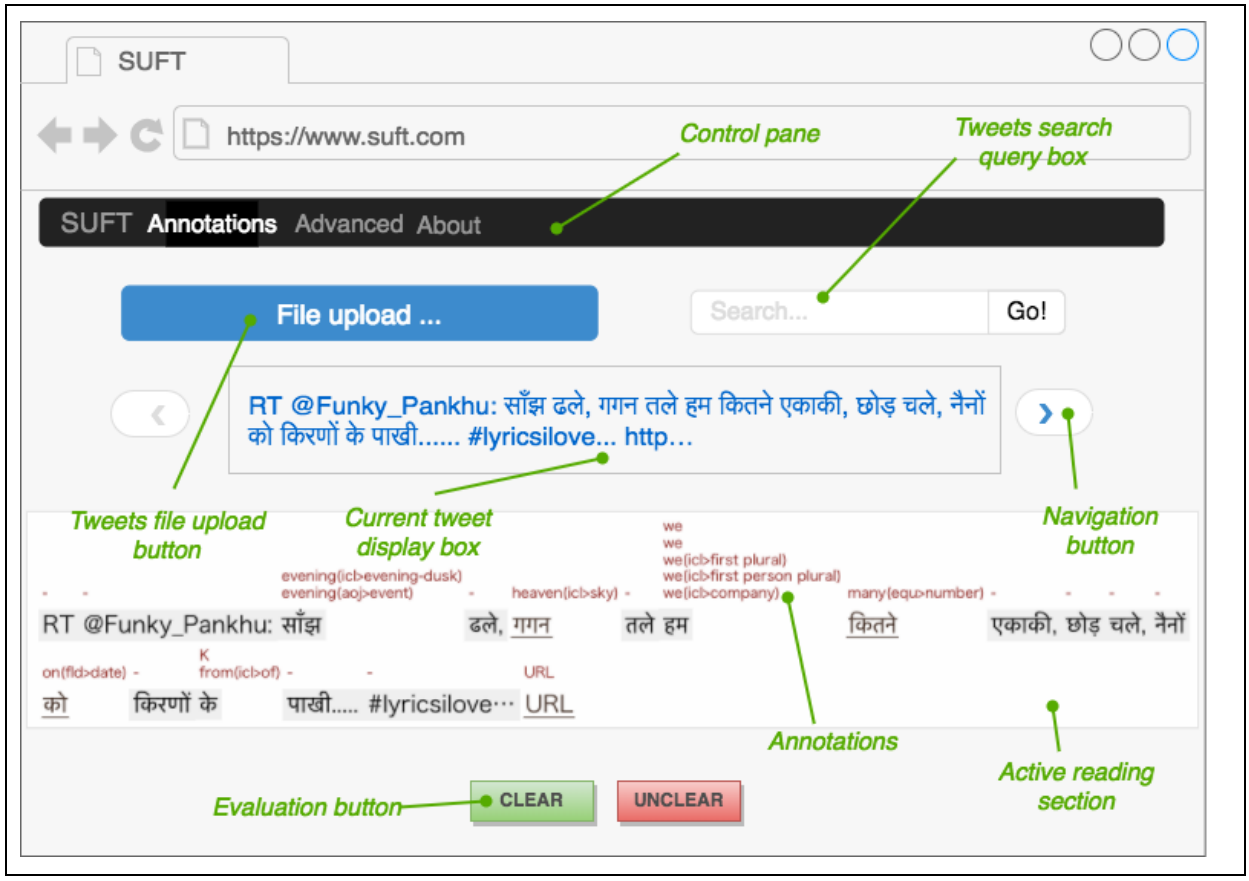

Figure 14: Screen display with annotations

II.1.1.1.2 Control pane

The SUFT-1 UI design is shown in Figure 14. Its control pane is placed horizontally at the top with several menus containing menu items for `Preferences`, `Language pair selection`, `Import` and `Export`.

The control pane might change in its layout in future versions and when adapted to mobile devices. In the control pane section, we also have a `Search` box to query for tweets and a `File upload` button to upload a set of tweets from a file. The loaded tweets can be navigated with the `Previous` (left arrow symbol) and `Next` (right arrow symbol) controls.

II.1.1.1.3 Annotated tweet part

We discuss some questions when deciding how to display tweet annotations. The decision choices are solely based on facilitating user comprehension.

In terms of the placement/layout of annotations a few pertinent questions are as follows.

1. Is it useful to have a horizontal layout or a vertical layout?
2. Should the pronunciations be placed above or below the tweet text?
3. Is lemmatized target annotation enough or would compatibility links be useful?
4. Should we plan to add "projected" properties (features) like Tense, Person, Gender, Number?
5. Is there some limit on the number of equivalences one should provide?
6. How to visually express relationships/links between disconnected lexical units?
7. Should we display any MT output alongside?

8. Where to display it, if any?
9. When to display it, if any? This could be based on some quality estimation (QE).

We also need to address the usefulness of dynamicity/interactivity of SUFT-1 and to determine whether the MT modules and SUFT-1 are exclusive or complementary.

For annotations, we plan to take compatibility links into account only if we want to test their usefulness, and possibly in later versions.

II.1.1.1.4 Evaluation section (buttons)

The evaluation section should contain easy and intuitive controls for evaluator feedback. The control should be preferably in a horizontal layout at the bottom.

Use of key bindings to the controls may prove to be more ergonomic during experimentation. This section will be made accessible only to evaluators of the system in later versions.

II.1.1.2 **Semantics**

II.1.1.2.1 Control pane

The control pane gives access to several menus to invoke various functions. The tweets are obtained and loaded using the `Search box` and the `File upload` controls.

As shown in Figure 14, the 'Search box' is used to enter queries (advanced operators of the TWITTER API can be used). Alternatively, the `File upload` control can help upload a local plain '.txt' file (with one tweet per line) for quick and offline experimentation. The loaded tweets will be displayed and browsable one by one by using the `Previous` (left arrow symbol) and `Next` (right arrow symbol) navigation buttons, as seen in Figure 14.

II.1.1.2.2 Annotated tweet part

As soon as one tweet appears in the control pane, the annotated tweet should simultaneously appear below. In an evaluation setting, this also triggers a timer which continues till the evaluator makes his input.

For creating the annotations, the system accesses dictionaries for the corresponding language pair, remotely or locally, and also refers to transcription modules. The transcription of each word is placed just above the word-form to facilitate readability for non-native users. Multiple kinds of annotations with varying degrees of detail may be displayed depending on the preferences. This is also true for any MT module which might or might not be displayed based on the user discretion.

In addition, the tweet may show static or dynamic annotations. For the dynamic case, clicking on annotations may show different behaviour in terms of color changes or order changes. Dynamic annotations may also be represented in the form of word lattices.

II.1.1.2.3 Evaluation section (buttons)

In an evaluation setting, the evaluator looks at the annotated tweet and determines its understandability. An internal timer records the time required by the evaluator to make his choice. Clicking on the understandable (Clear/Unclear) buttons (as in Figure 14) may then register an entry in the DB log along with the tweet, the choice made by the evaluator and the decision time. Later versions may include additional statistics.

II.1.1.3 API

We present now the APIs that can be used by the main SUFT-1 interface

`accept_upload (filename, dictionaryName)`

Table 14: Call on the server side to upload tweet file

API Name	`accept_upload`	Accepts two arguments
Argument 1 (mandatory)	`filename (type:String)`	the path of the file containing tweets (one tweet per line)
Argument 2 (mandatory)	`dictionaryName (type:String)`	name of the dictionary to access

`accept_search (query, dictionaryName)`

Table 15: Call on the server side to invoke query based tweet search

API Name	`accept_search`	Accepts two arguments
Argument 1 (mandatory)	`query (type:String)`	the search query (including advanced operators as allowed by TWITTER)
Argument 2 (mandatory)	`dictionaryName(type:String)`	name of the dictionary to access

II.1.2 Evaluation module

II.1.2.1 Visual Interface

II.1.2.1.1 *Screen display*

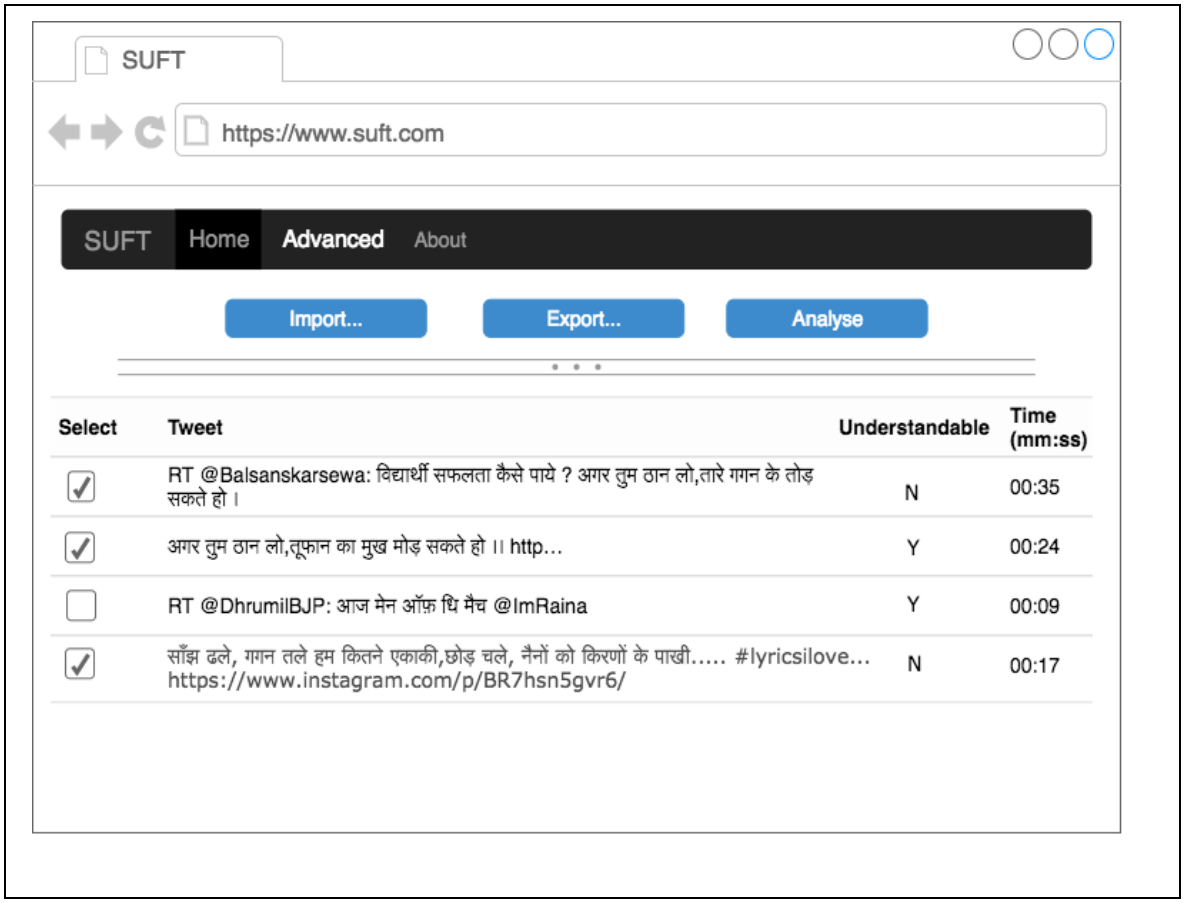

Figure 15: Screen with import/export functionality and selection controls for tweet logs

II.1.2.1.2 Configuring the log part

As the evaluator performs an experiment, the logs are stored in a database (MySQL). As shown in Figure 15, these logs can be imported so that the tweets can be viewed and selected for performing other operations. The tweets can be selected and the recorded observations can be used for making different kinds of analyses.

The log files can also be exported in various formats as desired by the evaluator (.csv for instance).

II.1.2.1.3 Parameters for generating statistics

The parameters to be recorded for later data analysis are as follows.

1. For each evaluator: proficiency level in the language(s) of the tweets, computer literacy, knowledge of the domain, interest in understanding the tweets, age, and degree of similarity between the language(s) of the tweets and his/her native and known languages. For instance, the word order of Indo-aryan languages is similar to that of Japanese, which may be a facilitating factor for an Indian wanting to understand Japanese tweets.
2. For each evaluated tweet: length, degree of code-mixing, understandability decision (binary), understandability decision time, and scenario.

For example, in a setting where the user has access to MT and AR (annotations), there are 3 possibilities:

a. Show first MT only, stop if decision is "understandable", display AR if not.
b. Show first AR only, stop if decision is "understandable", display MT if not.
c. Begin with MT and AR both displayed.

In cases a and b, the time taken by each of the 2 steps should be recorded.[49]

3. For each evaluated tweet: SL-TL lexical coverage (% of words in the tweet that have at least one translation in the AR display), and if possible QE[50] of the available MT output(s).

II.1.2.2 **Semantics**

II.1.2.2.1 Tweets management for evaluation

In evaluation sessions concerning several evaluators and closed sets, we have different strategies:

1. Give all tweets of the test set in the same order to each evaluator.
2. Give all tweets of the test set in a random order to each evaluator.
3. Distribute the tweets (from a larger set) to the evaluators in a way to ensure a fixed repetition rate. For instance, each evaluator should evaluate 100 tweets, 20 of them are also evaluated with exactly 1 other evaluator, and 10 by exactly 2 other evaluators.

Only the first 2 strategies have been used so far.

[49] This is not yet implemented in SUFT-1.
[50] *A priori* quality estimation.

II.1.2.2.2 Content of logs according to parameters

Parameters for later analyses:

1. Understandability of the tweet: Yes/No.
2. Understandability decision time of the tweet: mm:ss (mm~minutes, ss~seconds).
3. Proportion (%) of annotations per tweet (excluding URLs, tweet metadata): [0-100].

Other significant factors include the language pair (tweet source language and annotation target language), ID of the evaluator (hence, the profile), proportion of annotated words per tweet.

II.1.2.2.3 Statistics parameters and possible values Parameters for later analyses:

1. Understandability of the tweet: Yes/No
2. Understandability time of the tweet: mm:ss (mm~minues, ss~seconds)
3. Proportion (%) of annotations per tweet (excluding URLs, tweet metadata): [0-100]

Other significant factors include the language pair (tweet source language and annotation target language), ID of the evaluator (hence, the profile), proportion of annotated words per tweet.

II.1.2.3 **API**

The APIs used within the evaluation module are outlined as follows.

`POST "#" data:{tweetID, evalTime, understandability, coverage, moduleInformation, userID}`

Table 16: Call on client side for posting observed variables to the server side

URL	`'#' (root URL)`	
Method	`POST`	
URL parameters	`tweetID (type:Integer)`	ID of the tweet assigned by SUFT-1
	`evalTime (type:Integer)`	evaluation time in seconds
	`understandability (type: Boolean)`	understandability decision (True/False)
	`coverage (type: Integer[1-100])`	indicates proportion of annotated words per tweet
	`moduleInformation (type: Integer)`	contains a sequence of 0s and 1s; the n^{th} position if assigned to a 'Yandex Translation module' will determine whether it was enabled during evalution depending on its value (0~disabled)
	`userID`	ID of the user/evaluator to store profile

`logEvaluation(tweetID, evalTime, understandability, coverage, moduleInformation, userID)`

Table 17: Call on server side for writing observed variables to the database

API Name	logEvaluation	Accepts six arguments
Arguments	`{tweetID, evalTime, understandability, coverage, moduleInformation, userID}`	Same as URL parameters of Table 16

II.1.3 Dictionary management module

II.1.3.1 Visual Interface

II.1.3.1.1 *Screen display*

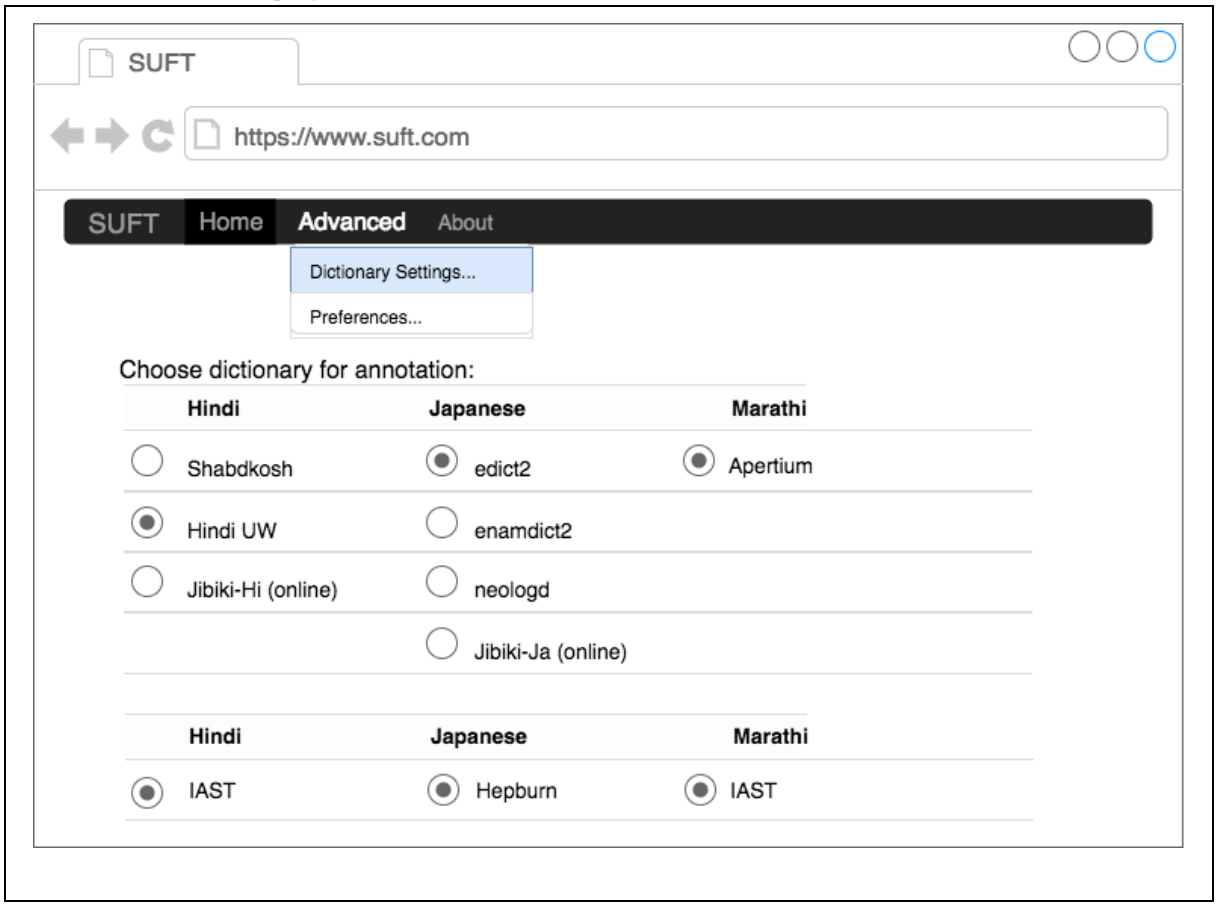

Figure 16: Screen for the multiple dictionary selection

II.1.3.1.2 *Dictionary settings menu*

The 'Dictionary Settings' menu item can be accessed from the 'Advanced' menu. As shown in Figure 16, clicking on it brings up a display containing two sections.

The first section shows a list of bilingual online/offline dictionaries available for each source language handled by SUFT-1. These will be used for annotating the tweets.

The second section allows users to select a transliteration scheme for each language.

II.1.3.1.3 *Selection controls for dictionary*

As seen in Figure 16, the dictionary names and transliteration schemes for each language are arranged column-wise.

The labels of all options are accompanied by radio-buttons so that only one dictionary per language can be selected by the user. A dictionary label indicates if the dictionary is a local resource or an online resource.

II.1.3.2 **Semantics**

II.1.3.2.1 Data preparation for dictionary

As an integral part of SUFT-1, resources in terms of differently enriched bilingual dictionaries are prepared from various sources or identified as available services for online access. The dictionary import procedures are prepared by the developers based on the various system constraints.

Quick and efficient dictionary lookup (online/offline) imposes several performance constraints and for better vocabulary coverage lemma-based lookup requires morphological analyzer integration. The dictionary data design, preparation and management is done by developers.

II.1.3.2.2 Data and dictionary management by developers

Developers are responsible for data preparation and the software implementation for integrating the local and online dictionaries. Dictionary search with morphological analysis requires data to be prepared in various formats (e.g. formats for ATEF).

A mechanism to load mini-dictionaries in main memory is introduced to improve efficiency. The software developers consider these factors when adding a new dictionary to the existing set and make suitable changes to the implementation.

Multiple transliteration schemes can be added to the software by the developers. The users or evaluators of SUFT-1 use the "Dictionary Settings" shown in Figure 16 to select a particular dictionary or a particular transliteration scheme as desired.

II.1.3.2.3 Dictionary selection and use by evaluators

Evaluators can select which dictionaries are used (in each experimental context), and later assess their usefulness and coverage.

Note that a dictionary D1 having less coverage than another dictionary D2 can be more useful to some user because, for instance, it covers more words that are ignored by the user.

II.1.3.3 **API**

The API calls related to dictionary access are as follows.

`get_senses_from_onlineDict(tweetWords, dictionaryName)`

Table 18: Call on server side to obtain senses from a dictionary available online

API Name	`get_senses_from_onlineDict`	Accepts two arguments
Argument 1 (mandatory)	`tweetWords (type:Array[String])`	words of the tweet to be annotated
Argument 2 (mandatory)	`dictionaryName (type:String)`	name of the dictionary to access

`get_senses_from_localDict(tweetWords, dictionaryName)`

Table 19: Call on server side to obtain senses from a dictionary on the local file system

API Name	`get_senses_from_localDict`	Accepts two arguments
Argument 1 (mandatory)	`tweetWords (type:Array[String])`	the tweet text to be annotated
Argument 2 (mandatory)	`dictionaryName (type:String)`	name of the dictionary to access

II.1.4 SUFT-1 Controller module

II.1.4.1 Functional diagram

II.1.4.1.1 SUFT-1 Controller and external components

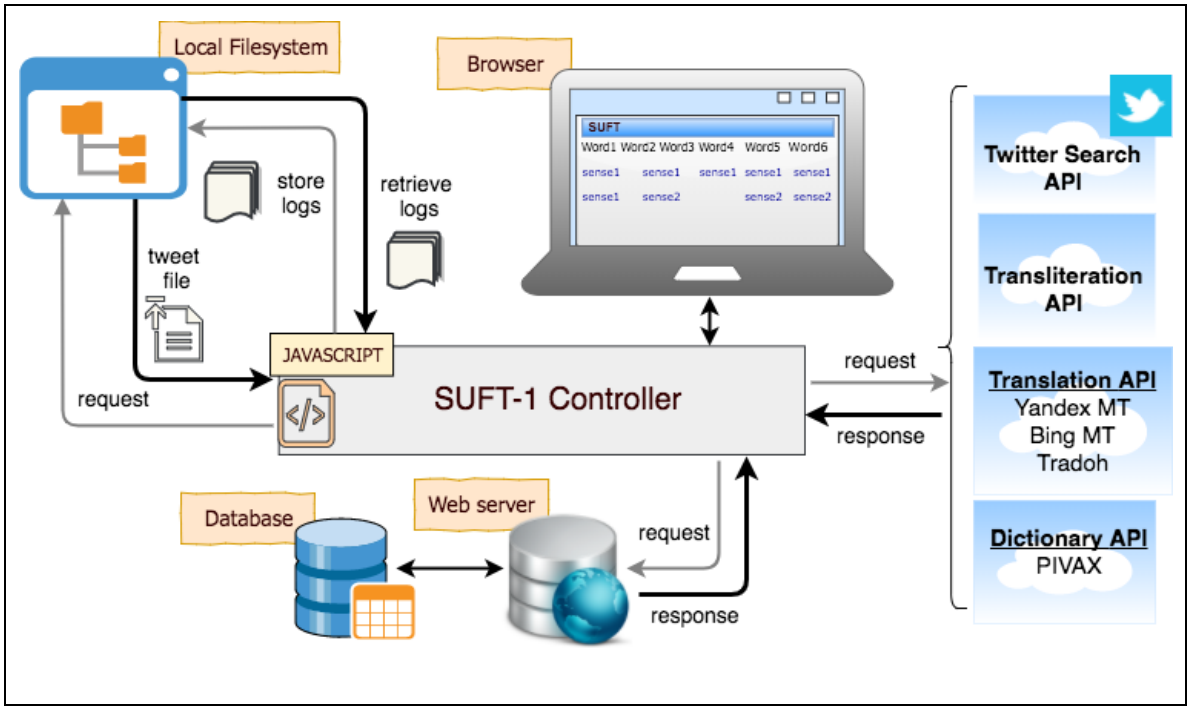

Figure 17: Interaction between the SUFT-1 Controller module and external components

II.1.4.1.2 Description

As shown in Figure 17, the SUFT-1 Controller module controls the interactions with all the required external components.

1. The module controls the mechanisms for inputting tweets either through a file upload, online search API access or through DB logs for analyses of evaluated data.
2. It manages the granularity and type of annotations provided alongside the tweet, thus controlling the output.
3. The controller contains several sub-modules which are specifically implemented to load resources in the form of offline or online dictionaries
4. Optional translation services are accessed and managed by means of APIs by the module.
5. Interactions with miscellaneous modules like the ones providing transliteration for various languages are also managed by the SUFT-1 Controller.
6. Under the evaluation setting too, the storing and retrieval of logs in a DB or otherwise is handled by the module.

II.1.4.2 Semantics

The interaction between several components, as seen in the earlier section, is handled by the SUFT-1 Controller module. The SUFT-1 system could accept as input, multilingual tweets by means of a file upload mechanism for quick experimentations. The file is expected to be a plain text utf-8 encoded file with one tweet per line. For on-demand search-based tweets, SUFT-1 accesses the TWITTER search API and the functionality is implemented as a part of the SUFT-1 controller module. The JSON responses from TWITTER are parsed to extract text

statuses. The tweets inputted from either mechanisms are loaded and displayed one-by-one while simultaneously initiating a process of dictionary lookups.

The dictionary lookups are either direct or routed through the lemmatisation process and are handled appropriately for online/offline situations. The information obtained from the user-selected dictionary for a particular language is loaded on successful lookups. These data structures are then utilised according to the required detail of annotation and displayed.

Translation services available through open-source APIs, as well as in-house MT systems, are managed by the SUFT-1 controller. The controller manages the API calls with the necessary information (language pair and direction etc.). This applies also to the transliteration module depending on the choice of scheme and language involved. The transliterations show up as labels and are placed above each word in the tweet.

In an evaluation setting, the SUFT-1 controller manages the mechanism of preparing the logs with appropriate information and statistics evaluated during the experiment and then writes it to either a file or DB for further analyses. The retrieval of logs for estimating various parameters is handled by the controller as well.

II.1.4.3 **API**

We show here the API for accessing the Yandex MT output from the server

```
getYandexTranslation (source_text, srcLang, tgtLang)
```

Table 20: Call to the server side for obtaining translation from Yandex MT

API Name	`getYandexTranslation`	Accepts three arguments
Argument 1 (mandatory)	`source_text (type:String)`	tweet text to be translated
Argument 2 (mandatory)	`srcLang (type:String)`	name of the source language
Argument 3 (mandatory)	`tgtLang (type:String)`	name of the target language

II.2 Internal specifications

II.2.1 Screens design

II.2.1.1 Rationale for the choice of UIKIT

We have decided to use UIKIT[51], a front-end framework for developing interfaces because

1. it is fast and powerful,
2. it offers a very rich set of web controls, and
3. it is geared towards being lightweight and modular (this is useful in the long term as we plan to port SUFT-1 on tablets and smartphones).

II.2.1.2 Technical specifications

For quick file uploading, the desired input format could be plain text files with one tweet per line. Alternatively, a client-side javascript implementation allows searching tweets through the TWITTER REST API.

The tweets are loaded alongwith simultaneous annotations asynchronously in the background but are displayed as the tweet collection is navigated one by one. Transliteration above each word is obtained through available javascript libraries. The dictionary translations or other

[51] https://getuikit.com/docs/introduction

information below the word is based on lemma-based or direct lookups in the selected dictionaries.

A translation module within a foldable container-like web-component can be hidden or shown at discretion. This displays the translation of the tweet using some automatic MT service supporting the language pair concerned. For consistency, the layout is the same across different source languages and allows different display sizes.

II.2.1.3 Other specifications

The translation modules as a complementary aid to the annotations can use services like TRADOH, YANDEX TRANSLATE API or BING TRANSLATE API. The framework used, here UIKIT, facilitates the modification of layout as and when desired.

II.2.2 Task controller module

II.2.2.1 Necessity

The SUFT-1 task controller is responsible for handling interactions with the various components in a multilingual setting. It is therefore necessary for the controller to use techniques which make the end-to-end process efficient and use techniques to that effect.

In particular, the task controller implementation will be based on the internal specifications for online communications with API calls, dictionary loading and lookups which then have to be laid out very carefully.

Specifications for the SUFT-1 task controller are driven by various factors like the kind of resources, access mode, tool capabilities, size, scalability and performance constraints. It is eventually desirable to select a framework which facilitates the implementation of such specifications.

II.2.2.2 Available technique/tools

To implement the SUFT-1 task controller, parameterised exchanges over HTTP maybe sufficient without the use of distributed messaging services like ACTIVEMQ. The SUFT-1 controller needs to load various online services like calls to TWITTER API for tweets and the Hindi MA, PIVAX-3 and PAPILLON web services for dictionary accesses.

A large typical lexical dictionary, which occupies about 1GB of disk space for, say, 1.6 million entries, has to be loaded in some form (e.g. the form already compiled for ATEF). It is important that the dictionary size does not exceed 1MB (for portable devices) and has a small memory footprint in any case. This necessitates methods or solutions to load dictionaries from zip files as opposed to storing entire DOM structures, which is expensive as seen from earlier tools like PIVAX[52].

Another possible solutions could be to use hash databases like KYOTOCABINET, with provide very efficient read accesses, even for very large dictionaries. Yet another specific solution would be to call PIVAX to obtain a mini-dictionary of 100K words at the cost of losing dynamicity probably.

Using a framework like CORDOVA (PhoneGap) or UIKIT, coupled with programming languages such as JS and PHP for implementing web applications, is the best current solution to make the system scalable and portable. These frameworks also support flexible layouts.

[52] PIVAX-3, based on Jibiki-2, is much faster (Ying, 2016), so we might try it in the future.

II.2.2.3 Choice and description

For initial versions of SUFT-1, we use technologies similar to the ones used in the PAPILLON[53] project (Boitet, Mangeot, & Sérasset, 2002), which has built a multilingual lexical database of about 2M entries coming from contributed dictionaries in 9 languages since 2001. The Papillon web service allows open access through a REST API for dictionary lookups using lemma-based matches. TREETAGGER and MECAB are used as morphological analyzers for processing French and Japanese texts respectively, and then the lemmas are used to retrieve entries.

On the client side, the front-end technologies include JS and HTML, while, on the server side, PHP and MYSQL databases are used. AJAX services are employed to facilitate minimal and lightweight interactions with the databases. The API provided for dictionary access uses an XML request and response mechanism.

II.2.3 Target language graph

II.2.3.1 Graphs produced by typical morphological analyzers

Classically, an input to a morphological analyzer (MA) is a sequence of word forms (typographical words in the writing systems having word separators like spaces and punctuations) denoted by $\Phi = \phi_1\phi_2 \ldots \phi_n$.

Φ can be represented by a graph in 2 ways, chart and lattice.

II.2.3.1.1 Chart

Like in finite-state automata, the information is on the arcs.

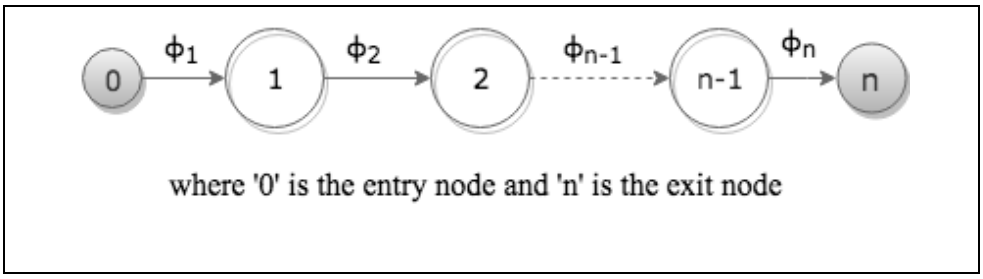

Figure 18: Chart representation of a word form sequence Φ

II.2.3.1.2 Lattice

Here, the information is on the nodes, and the arcs indicate compatibility at distance 1.

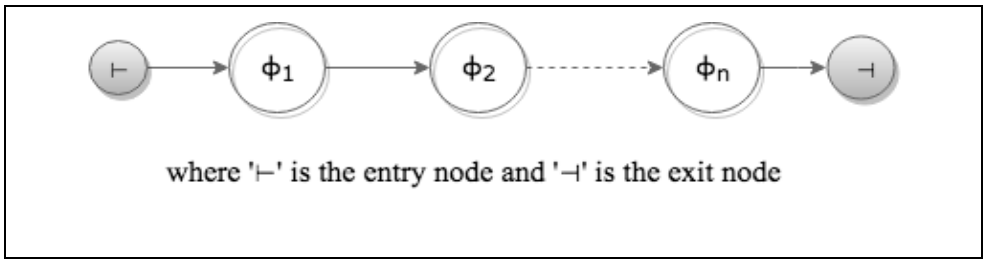

Figure 19: Lattice representation of a word form sequence Φ

[53] http://papillon.imag.fr/papillon/Home.po

II.2.3.1.3 Results as charts

Tools based on Finite-State Transducers (FSTs) such as NooJ usually adopt the chart representation, and interpret it as a FSA. Analysis "intersects" it with the FST representing the grammar and produces another FSA that is no more linear in general.

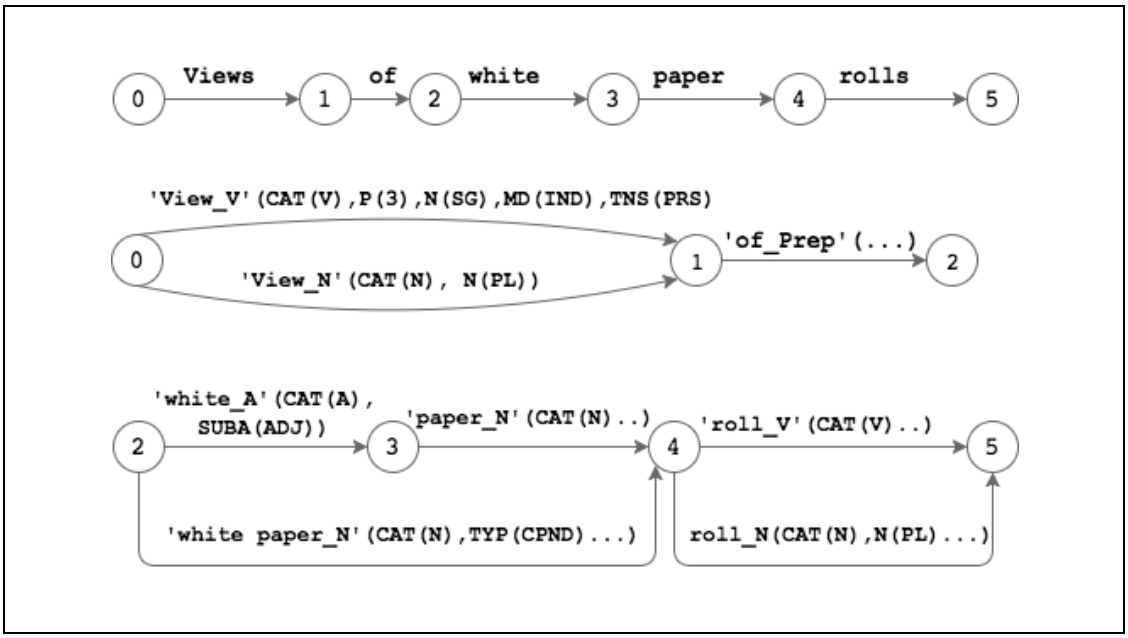

Figure 20: Example of an FSA representation produced after analysis

II.2.3.1.4 Results as lattices

ATEF-78 had this sort of output and ATEF-Y will also offer it. However, the outputs in the form of decorated trees are often preferred, and such a tree is similar to a lattice. Indeed, to convert a forest into a lattice, one just adds an entry node '⊢' above the roots, and an exit node '⊣' under the leaves. The lattice form allows to express compatibility constraints without having to duplicate the analysis of any word (Seligman, Boitet, & Meddeb-Hamrouni, 1998). Here, we suppose that a verb cannot follow the compound noun 'white paper'. In the graph produced by MA results are compatible if they are on one path from the entry node to the exit node.

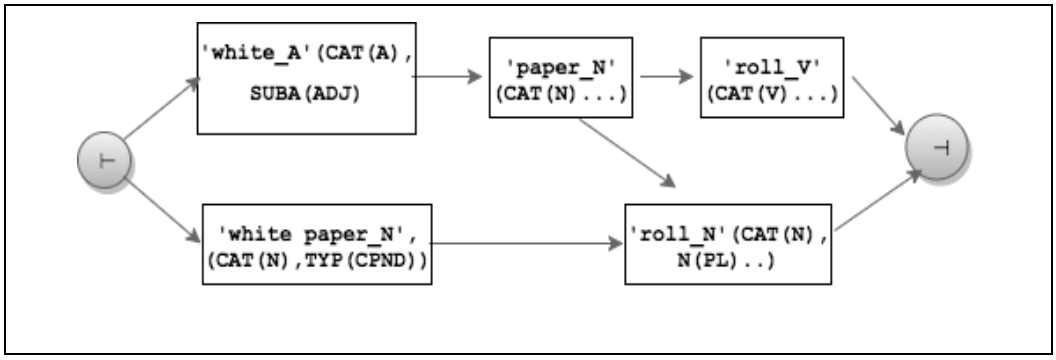

Figure 21: Representation of the analysis as a lattice

II.2.3.2 Handling the graph data

We present a simple morphological analyzer written in ATEF in Section II.2.4.1. It returns (possibly) several analyses[54] for each *occurrence*[55], based on the content of the dictionaries provided as ATEF *lingware components*.

SUFT-1 expects then data from ATEF, for each word form with various analyses in a data structure capable of being processed as a graph and which can be transformed in a visual representation to be shown along with the annotations.

II.2.3.3 Visual representation of the ATEF output in SUFT-1

```
'अंधे': UL('अंधा'), CAT(J), SUBJ(ADJ), GEN(MAS), NUM(SNG), CAS(OBL).
'अंधे': UL('अंधा'), CAT(J), SUBJ(ADJ), GEN(MAS), NUM(PLR), CAS(DIR).
'अंधे': UL('अंधा'), CAT(J), SUBJ(ADJ), GEN(MAS), NUM(PLR, CAS(OBL.
'अंधे': UL('अंधा'), CAT(N), GEN(MAS), NUM(PLR), PER(TD), CAS(DIR).
'अंधे': UL('अंधा'), CAT(N), GEN(MAS), NUM(SNG), PER(TD), CAS(OBL).
```

Figure 22: Output produced by ATEF for Hindi word-form 'अंधे' (blind) with multiple analyses

Unassigned attribute, like in this example `TNS` (tense) or `MOOD`, are not printed out by default. That is equivalent with writing `TNS(TNS0)` and `MOOD(MOOD0)`.

II.2.4 Embedded morphological analyzers

II.2.4.1 ATEF-based MAs

ATEF is a SLLP (specialized language for linguistic programming) used to build morphological analyzers. MAs of very large coverage and exactness have been written for Russian, German, English, French, Portuguese, and more recently Lithuanian and Spanish. In 1974, it has been used even on Japanese, handling a sentence (with no spaces) as a long compound word (PhD by Annick Laurent).

The underlying computing model is an extended FST capable of handling the three levels of morphological analysis, namely flexional, derivational and compositional (Chauché, 1975). We plan to use ATEF to build a truly multilingual morphological analyzer for Indian languages and English, because of the high degree of code-mixing in Indian tweets.

For that, we need to prepare a large morphological resource in the format required for ATEF. Note that ATEF allows to easily build powerful grammars which can recognize usual OOV words as compound words. It is also possible to write a sophisticated subgrammar to handle out-of-vocabulary words: for example, '-ations' can be stored as a possible ending for action nouns in plural deriving from regular verbs, so that, given the word form 'tweetizations', the ouput could be the hypothetical lemma `'tweetize'`, with the attribute list `DERIV(Verb2actionN), NBR(plural), SEM(ACTION).`

II.2.4.2 MeCab MA for Japanese

As the current version of ATEF cannot segment a Japanese sentence into words, using ATEF for the MA of Japanese tweets would require a segmentation processing. But it is a better idea to use MeCab[56], that performs both segmentation and morphological analysis, is open source,

[54] An "analysis" is a *vector* called in ATEF *decoration,* which contains the surface string, the lexical reference — lemma, or derivational family), and values of various morphosyntactic attributes such as POS, number…).
[55] In ATEF, that means a maximal character string not containing a space (nor a newline of EOF character). Occurrences are in general word forms, but html tags, numerical dates, etc., are also occurences.
[56] http://taku910.github.io/mecab/

and has a large coverage. The PAPILLON web service (Mangeot-Nagata, 2016) as well as the Cesselin jp-fr AR tool use MECAB.

The system can be locally installed and accessed by SUFT-1 using wrappers for command-line invocations from a web programming language (e.g. PHP). Recent versions of MECAB return UTF-8 encoded output and are also available with multiple script bindings in case required. Our plan for Japanese is hence to use the desired level of information from the MECAB output and to project those grammatical symbols into English or French as necessary.

II.2.4.3 **Interface for other lemmatizers**

Lemmas are useful to access dictionaries and preliminary level of content extraction. LEXTOH (Ying, 2016)is a lemmatization middleware for calling one or more morphological (and possibly morphotactic) processing tools on a raw text or an inputted formatted document, and to produce a result containing at least the lemmas, in a certain formalism. It is also possible to employ it for searching in a computerized dictionary or a lexical database.

The LEXTOH service is provided as a REST API. We plan to use it in the future to use existing MAs for other languages for which no ATEF MA has yet been developed and for "normalizing" their results into our format.

II.3 Interesting aspects of the implementation

II.3.1 **Tweet-related programs**

II.3.1.1 **Programs for extracting and filtering multilingual tweets**

II.3.1.1.1 Objectives and approach to the implementation of Tweezer

In order to make a study on the usage of tweets and to determine various statistics concerning the vocabulary of tweets, code-mixing proportions, number of unique tweets and other metadata, we wanted to implement specific programs. The objective was to use solutions that allowed us to access tweet data primarily through the REST APIs and the Streaming APIs, and to easily handle UTF8-encoded text for detailed analyses.

We used the IPYTHON NOTEBOOK interactive environment and gradually built a large code base for analysing tweets as and when required. We call this code-base TWEEZER (TWEEt analyZER). The programs have been implemented in Python and use various text processing libraries provided with the Python package. For the large quantities of multilingual tweets collected using the TWITTER APIs, we developed some custom features such as maintaining metadata logs.

II.3.1.1.2 Custom features for management and analysis of multilingual tweets

The program maintains logs in plain text files containing information about the query used for the extracted set, the quantity of tweets, the absolute path of the JSON file (filename includes the date/time of extraction). Example of a log entry:

```
{lang:hi OR ("indian music" OR "indian festival" OR "indian restaurant")} :1000
:/Users/riteshshah/mRECSYS/data/json/qhi/tweets-1000-01-07-2015-1535.json
```

This is useful for organising the tweets and getting a summarised view. Files are stored in appropriate directories specific to each language. The program also maintains tweet TXT files converted from their JSON counterparts for easy command line interface (CLI) processing whenever required. Program modules which concern only text processing necessarily load only the TXT instead of JSON and generate the statistics.

Some interesting statistics concerning Hindi tweets analyzed by our program have been tabulated ahead.

II.3.1.1.3 Analysis and observations for a collection of Hindi tweets

Table 21: Extraction of Hindi tweets using basic querying

Query set	#Extracted tweets	#Unique tweets	#Vocabulary	#Filtered (F1)	#ASCII (F2)	F2/F1*100(%)
Q1	78182	29076	95466	69236	3158	4.56%
Q1+Q2	84630	32239	106165	76262	9702	12.72%

In Table 21, #Filtered terms are terms from the vocabulary after removal of 'RT', URLs, hashtags and usernames; ASCII terms are a subset of #Filtered terms containing only ASCII charset; Q1 is the query set *{"lang:hi"}* and 'Q2' is the query set *{'lang:hi OR "indian music" OR "indian festival" OR "indian restaurant"'}*. The collection of Hindi tweets was obtained by submitting query Q1 (11 times) and query Q2 (twice) at different times. Based on the data in Table 21, we can make the following remarks.

1. Considering that the API allows a maximum of 18000 tweets per query, 78182 tweets (using Q1) is much less than (18000*11). This could mean that either less Hindi tweets were generated between successive Q1 submissions, or this could be simply attributed to limitations of the TWITTER search API vis-à-vis the submitted queries.
2. Filtering out duplicate tweets decreases the tweet count by less than 50%.
3. Also, in terms of code-mixing within the tweets, we see that tweets extracted using Q1 show about 4.56% of code-mixing, but this percentage jumps to 12.72% with an addition of only 6448 tweets (3163 unique) obtained using Q2, which accounts for an increase of 9.7% in the tweets.

II.3.1.2 **Use of query formulation techniques for obtaining relevant tweets**

II.3.1.2.1 Approach

Based on the observations made in the previous section, we performed a short experiment to examine whether query terms written by native speakers of the language of the tweets help maximise the collection of Devanagari scripted Hindi tweets. We systematically developed a set of semantically related query terms (from tourism domain) using the synsets of Hindi Wordnet, and then successively submitted each of them to the TWITTER API.

We describe the steps of the procedure used to obtain these query terms.

II.3.1.2.2 Procedure and results

Step 1: We selected a set 'TSWD' of native scripted tourism-related Hindi words: {'रेस्तराँ' (restaurant), 'लॉज' (lodge), 'उत्सव' (festival)}'.

Step 2: We then retrieved senses from the Hindi Wordnet by querying with each word in TSWD.

Step 3: We selected synsets belonging to a required sense (tourist-related, useful to tourists) and collected the synset words.

Step 4: For each synset then, we recursively went through the hyponyms and collected their synset words.

A total of 93 terms [cf. Appendix 1] was generated from the above steps.

Remark: A larger set of terms can be programmatically collected by traversing elements belonging to a synset closure of the seed query term.

The search API returned 31665 tweets in total when queried using the above terms.

Out of the 31665 retrieved tweets, the count of unique tweet messages stood at just 8902 (i.e. 28%) with a vocabulary of 32588. The code-mixing percentage however, drops down to 1.5% as can be seen from Table 22 below. #Filtered terms are terms from the vocabulary after removal of 'RT', URLs, hashtags and user names; ASCII terms are a subset of #Filtered terms containing only ASCII characters.

Table 22: Tweet extraction using Devanagari scripted terms

Query set	#Extracted tweets	#Unique tweets	#Vocabulary	#Filtered (F1)	#ASCII (F2)	F2/F1*100(%)
TSWD	31665	8902	32588	23789	348	1.5%

II.3.2 Algorithm for graph presentation and manipulation

II.3.2.1 Use of graph description languages and rendering libraries

The MA of a tweet text produces a lattice of all possible analyses. In addition to the display of the annotations (pronunciation, meanings), it might be useful to represent this lattice in a visual form. This can be done for instance, by using GRAPHML, DOT, GRAPHVIZ, CANVIZ & GEPHI, which convert such information to a visual representation.

The programs mentioned above take descriptions of graphs in a simple text language and transform them into visual representations. For example, GRAPHVIZ[57] allows making diagrams in useful formats, such as images and SVG for web pages; PDF or POSTSCRIPT for inclusion in other documents; or display in an interactive graph browser. GRAPHVIZ has many useful features for concrete diagrams, such as options for colors, fonts, tabular node layouts, line styles, hyperlinks, and custom shapes.

There are also some javascript libraries, for example CYTOSCAPE[58] that allows interaction with the nodes of the representation.

In SUFT-1, we do not try to represent the morphosyntactic compatibility lattice produced by a MA in an exact way. We simply show its approximation as a confusion graph.

II.3.2.2 Interactive functions

The users can manually interact with the graph nodes. A similar interactive functionality could be added by using javascript and HTML for choosing word senses available as annotations to each tweet word. The user can 'push up' or 'push down' some senses depending on the context.

[57] http://www.graphviz.org/
[58] http://js.cytoscape.org/demos/multiple-instances/

In our horizontal presentations, the possible equivalents for a word (word form) are shown vertically, as a list allowing to push up or push down an item. In the vertical presentations (like that of laosoftware.com), we simply allow the user to highlight one equivalent among the possible equivalents of a word, because we found no widget allowing to move an element of the list of equivalents left or right.

II.3.3 Efficient memory processing

II.3.3.1 Mechanisms to load a large collection of mini-dictionaries in main memory

When the SUFT-1 controller attempts to annotate the segments of the tweet text, it consults the appropriate dictionary or lingware for annotations or/and set of possible analyses of simple words, compound words or idiomatic expressions etc. However, when consulting dictionaries especially, in online mode, it is important to look at the efficiency aspect.

For example, simply loading DOM structures or entire dictionaries is computationally expensive and hammers the response time of the annotations. PHP provides mechanisms to load and buffer files, however, it imposes a limit of 30MB in such a case.

We therefore resort to alternative solutions like loading mini-dictionaries in main memory using services like CREATDICO (Ying, 2016) that make it possible to produce the dictionary for each lemma in a generic way. The solution presented in (Huynh, 2010) describes the storage of a mini-dictionary associated with a segment in the database, so that they are prepared in advance and always available. The mini-dictionary technique is a kind of a proactive help that can be produced for each tweet, and thereby reduces the memory footprint with a faster response time for annotations.

Another solution for quick accesses, especially dictionary look-ups in dictionaries stored locally, is the KYOTOCABINET[59] (KC). KC is a kind of a hash database where the records are organized in a hash table or as a B+ tree. Each operation of the hash database has an O(1) complexity. In theory, performance is unchanged when size increases. However, in practice, performance is determined by memory speed or storage device speed. The upper limit of the DB size is 8 EB (exabytes), however, even if size exceeds main memory capacity, a maximum of 1 or 2 seeks of the storage device is required. Additionally, preprocessing is required on inconsistently prepared raw dictionaries in order to normalize them and transform them into a .TSV format for compiling in KC.

II.3.3.2 Other details

The multilingual tweets obtained from various resources have been processed using a software implemented in IPYTHON NOTEBOOK, a fast and scalable web environment used for programming and quick experiments. Our TWEEZER program (cf. II.3.1.1.1) has been gradually extended to extract multilingual tweets and store them in a language-dependent (Hindi, Marathi, Gujarati, Japanese) directory structure and perform various kinds of analyses on the acquired JSON files.

TWEEZER will be made available on the open-source public repository in BITBUCKET[60].

[59] http://fallabs.com/kyotocabinet/
[60] https://bitbucket.org/riteshms/tweezer

Synthesis

In this chapter, we presented the detailed external and internal specifications for SUFT-1, by taking into account all factors that will determine its usefulness in helping understand foreign tweets.

While discussing interesting aspects of our tweet-related programs, we presented experiments for Indian language tweet extraction that made use of various query formulations. We observed that this can help in increasing the recall of tweets with reduced code mixing.

Lastly, we discussed mechanisms for efficient memory processing, annotation graph presentation and manipulation to see how they can contribute to the ergonomy of the AR environment, which intrinsically determines the usefulness of SUFT-1.

In the next chapter, we describe the creation of a multilingual morphological analyzer specialized to tweets, an important resource for SUFT-1.

Chapter III Creation of a multilingual morphological analyzer for Indian tweets

Introduction

In this chapter, our goal is to build one large scale morphological analyzer accessible through a middleware to SUFT-1 because, to the best of our knowledge, there are no large-scale multilingual morphological analyzers suited to handling code-mixed Indian tweets.

We propose an approach to build a large coherent lexical database from several resources including simple or compound Indian tweet word forms. We discuss how we develop the other ATEF lingware components (variables, formats, grammar, dictionary) for handling noisy Indian language tweets. We conclude by evaluating and making remarks on the quality and coverage of the morphological analyses output and the underlying resource.

In Section 1, we present the goals for building a large scale MA, review the current state of the art and discuss the principles involved in building it. In Section 2, we describe how we coalesce data from several sources and build a large lexical database. In Section 3, we describe the methods used to generate ATEF dictionaries from this lexical database. In Section 4, we evaluate the quality and coverage of the MA resource and its output.

III.1 Goals, State of the Art, Principles

III.1.1 Goals

III.1.1.1 Construction of a large or very large resource

Tweets in one language (source) are annotated in another (target) by querying the dictionary "directly" by the word forms, or by the lemmas. Lemma-based lookup increases of lexical coverage.

III.1.1.2 Necessity of multiplicity of output and integration of language identification

Code-mixing is inherent in tweets, and therefore annotation of tweets necessitates multiplicity of output. Especially, for Indian language tweets where there is a possibility of homography in 1 language or sometimes in 2 languages (e.g. Hindi and Marathi), it is also important to integrate mechanisms of language identification.

We have noted earlier that, because English is the *lingua franca* of India, tweet texts are often code-mixed with English and tweet authors often use romanised transliterations, even when they express themselves in Indian languages. It is therefore important to consider Hindi, Marathi and English to go through morphological processing tools.

Here again, our solution is simple: we will precompute all forms of a large set of lemmas, for each language under consideration, and put the language name as value of one particular feature. If for example a string in devanagari can be a word form both in Hindi and in Marathi, the dictionary will contain that string twice, once with the information asociated to the Hindi word, and once for the Marathi word.

III.1.1.3 Ease of integration of several MAs in the future

Given the multilingual setting and the languages handled by SUFT-1, it is necessary to be able to use several morphological analyzers than to use a unique framework (like ATEF, NooJ, etc.) and use it to build a MA for each language.

Also, it may happen that there exist 2 or more MAs for a language, with different lexical coverages. In that case, we should aim at combining their results.

All this requires that several MAs can be integrated, that is, used at the same time by a SUFT. In SUFT-1, we use for the moment a weaker solution: we use ATEF for Indian tweets (and a unique MA handling Hindi and Marathi, soon also English, French and Gujarati), and MECAB for Japanese.

In the future, we plan to use like LEXTOH (Ying 2016), which is a lemmatisation middleware, to access as many Mas as possible, and combine their results in the available normalized attribute-value representations.

III.1.2 State of the art and proposed method

III.1.2.1 State of the art for large-scale & code-switched MAs

III.1.2.1.1 Large-scale morphological analyzers

In the multilingual context of SUFT, we have seen the relevance of integrated morphological analyzers for code-mixed tweets. In addition to simple word forms, it is equally desirable to be able to handle compound word forms, named entities, idiomatic expressions and OOVs, in a unified manner.

This necessitates mechanisms capable of merging different morphological analyses from the available tools and during analyses per se, it is also interesting to look at solutions based on precomputation, dynamic segmentation processes or FSTs such as ATEF.

To the best of our knowledge, there are no large-scale morphological analyzers specialised for code-mixed tweet-like texts.We review existing analyzers and find two morphological analyzers for Hindi (not especially suited to code-mixed texts) from IITB[61] and IIITH[62]. We propose strategies for building MA resources for ATEF in Hindi from existing Hindi analyzers and MA resources for ATEF in Marathi from existing dictionaries of lemmas and known paradigms.

III.1.2.1.2 Possible analysis strategies

a. Classical morphological analysis

Interesting references on Hindi morphological analysis include a FST based study (Bögel, Butt, Hautli, & Sulger, 2007) and a distributed MA approach (Singh & Sarma, 2011) and for Marathi, studies include (Bapat, Gune, & Bhattacharyya, 2010) and (Dabre, Amberkar, & Bhattacharyya, 2012), however none especially suited to noisy tweets. But (Sasano, Kurohashi, & Okumura, 2014) and (Saito, Sadamitsu, Asano, & Matsuo, 2014) describe investigations on Japanese morphological analysis concerning noisy text like tweets.

For SUFT-1, we propose building a morphological analysis resource based on segmentation and transformations of morphs (prefix, radical, affixes, suffixes) for the concerned Indian languages.

b. Pre-computation by generation

THE ETAP-3 system[63] has a generator and facilitates precomputation by using a « dictionary of word forms » in Russian with the associated MA results (lemma + POS and other features). In

[61] http://www.cfilt.iitb.ac.in/Tools.html from IITB (Indian Institute of Technology Bombay), India.
[62] http://sampark.iiit.ac.in/hindimorph/web/restapi.php/indic/morphclient from IIITH (Indian Institute of Information Technology Hyderabad).
[63] http://cl.iitp.ru/etap3

ETAP-3, the generator is capable of producing 4 million forms and produces all possible hypotheses in the form of a confusion graph or lattice.

We draw inspiration from ETAP-3 and the OMNIA project (Rouquet & Nguyen, 2009) where compound words are handled by making use of NooJ and the DELAF dictionary. However, unlike NooJ which is a pure non-extended FST, we plan to extend our methodology to using ATEF (Chauché, 1975) which is based on an extended non-deterministic finite-state transducer model.

c. Pre-computation by mix of methods

Essentially, our proposed methodology includes classical analysis complemented with manual or programmatic generation to complete the morphological "paradigms". For Hindi and Marathi, we gather resources in different ways, but transform them into a data structure to be used by ATEF.

This will be our preferred technique because we can use "pure data" and don't need to implement a morphological generator.

III.1.2.1.3 Handling code-switched tweets

To the best of our knowledge, there are no morphological analyzers specifically geared towards handling code-switched tweets (including OOV tokens, emoticons etc.) in Indian languages. In the context of SUFT-1, it is necessary to merge and integrate the various MAs for the languages at hand in a unified manner.

III.1.2.1.4 Handling OOV words

For handling OOV words, some attempts that have been made at using unsupervised approaches to identify morphemes (Virpioja, Smit, Grönroos, & Kurimo, 2013), (Krishn, Guha, & Mukherjee, 2012) could prove useful. Alternate solutions need to be investigated to handle unknown words for instance, generating edit-distance 1 candidates for better matches.

III.1.2.2 **Principles of the proposed method**

We propose a resource specialized to tweets, which produces a graph giving the various possible solutions (and segmentations) by making use of SLLPs (ATEF-1-TW for our case) and available resources and lingware. Example of unique or multiple MA results for Hindi and Marathi containing word forms and fixed idioms (pre-computed) are shown below.

III.1.2.2.1 Desired output for simple words

```
'कामनाएं':UL('कामना'), CAT(N), GEN(FEM), NUM(PLR), PER(TD),
CAS(DIR), TNS(TNS0), ASP(ASP0), MOOD(MOOD0).

'अरबियों':UL('अरबी'), CAT(N), GEN(MAS), NUM(PLR), PER(TD), CAS(OBL),
TNS(TNS0), ASP(ASP0), MOOD(MOOD0).
```

Figure 23: Example of ATEF desired results for Hindi simple word forms

```
'भक्ताला': UL('भक्त'), CAT(N), GEN(MAS), NUM(SNG), PER(TD),
CAS(ACC), TNS(TNS0), ASP(ASP0), MOOD(MOOD0).

'नद्यांनी': UL('नदी'), CAT(N), GEN(FEM), NUM(PLR), PER(TD),
CAS(INS), TNS(TNS0), ASP(ASP0), MOOD(MOOD0).
```

Figure 24: Example of ATEF desired results for Marathi simple word forms

III.1.2.2.2 Compound words and NEs

```
'बाल-बाल बचा': UL('बाल-बाल बचना'),  CAT(V), GEN(MAS), NUM(SNG),
PER(TD), TNS(PST),  ASP(PFT), MOOD(DCL).

'कुम्भ मेले': UL('कुम्भ मेला'), CAT(N), GEN(MAS), NUM(PLR), PER(TD),
CAS(OBL), TNS(TNS0), ASP(ASP0), MOOD(MOOD0).
```

Figure 25: Example of ATEF desired results for named entities and idioms

The first entry in Figure 25 contains an idiom in Hindi (बाल-बाल बचा) which means "to have a narrow escape" and the second is a named-entity (कुम्भ मेले) which is an annual festive event in India. For named entities and idioms (including compound words), we build the ATEF idiom dictionaries in the same way, containing the inflected forms if available.

III.1.2.2.3 Handling of OOV words

Taking into account the frequency of typing mistakes, we resort to a simple method. We propose to generate all possible candidates within a Levenshtein distance of one. These candidates could act as a fallback for words which fall under the typographical or the OOV category.

III.1.3 Prerequisites: ATEF lingware components for Indian languages

III.1.3.1 Components and their dependencies

As described in (Boitet, 1997), the ATEF phase contains two components of variables declaration (DVM, DVS), "morphological", "syntactic" and "general" formats (FTM, FTS, FTSG, the last one being optional), 1 to 7 grammars GRi (1≤i≤7), 1 to 6 dictionaries of "morphs" DICi (1≤i≤6), at least one of them being of "bases" (morphs with lexical references), and from 0 to 14 in Ariane-G5 dictionaries of fixed connected idioms[64], DICTi (7≤i≤20). The dependencies between these components are shown in Figure 26.

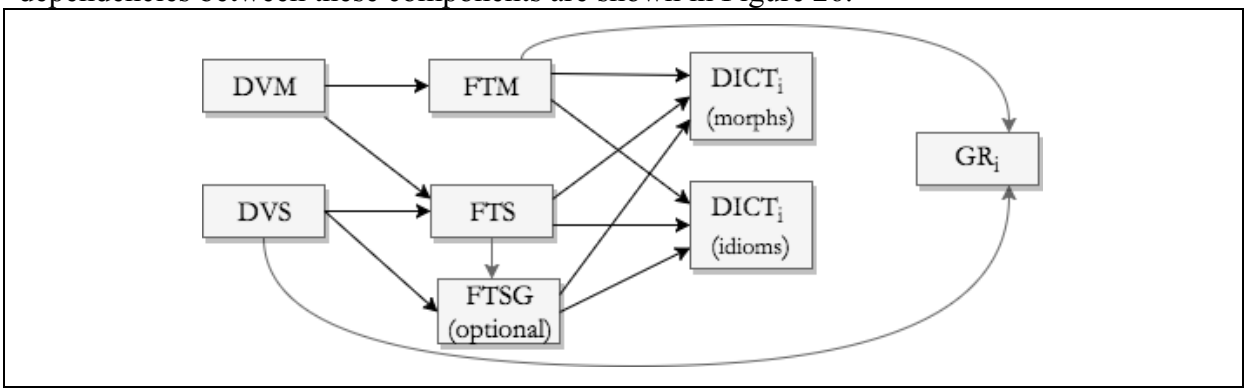

Figure 26: Dependencies between the ATEF lingware components

III.1.3.2 Fixed components

III.1.3.2.1 Variables declaration components (DVM, DVS)

We create two variable declaration files (components) of ATEF; DVM (morphological) as shown in Figure 27 and DVS (syntactic) as shown in Figure 28, for an MA phase aiming at handling tweets in Indian languages (Hindi, Marathi) and English.

[64] "Tournures figées connexes" in French (« Fixed Contiguous Idioms » in English), hence the T in DICTi.

Excerpts of the format of the two components are shown in Figure 29 and Figure 30 with details in Appendix 3.

```
-EXC-
** Subcategory of Nouns.
SUBN  == (C,    ** Common.
          P,    ** Proper.
          V,    ** Verbal.
          ST    ** Spatio-temporal.
         ).
```

Figure 27: Excerpt of a syntactic variable declaration file

```
-EXC-
** Languages: EN-English, FR-French, HI-Hindi,
   MR-Marathi, GU-Gujarati, XML-tags, EMO-emojis.
LANG == (EN, FR, HI, MR, GU, XML, EMO).
```

Figure 28: Excerpt of a morphological variable declaration file

III.1.3.2.2 Morphological and syntactical formats (FTM, FTS)

```
FTMHIW == LANG-E-HI.
FTMMRW == LANG-E-MR.
FTMENW == LANG-E-EN.
FTMFRW == LANG-E-FR.
FTMGJW == LANG-E-GJ.
FTMXML == LANG-E-XML.
FTMEMO == LANG-E-EMO.

MODINC == CAT-E-UNK.
```

Figure 29: Excerpt of a morphological format file

```
FTSEMPTY ==. ** We need only 1 FTS format.
```

Figure 30: Excerpt of a syntactic format file

III.1.3.2.3 Grammar

The following very simple grammar showin in Figure 31 produces in 1 step the information attached to a word form in the dictionary. RDICT initializes the analysis of a word form by opening the available dictionaries. In this case, all word forms are in D4. Their language is indicated in their FTM (morphological format). If the word form ϕ to be analysed is present in the dictionary, the rule RWORD applies, for each article indexed by ϕ. Execution is non-deterministic all-path with backtrack, so that all solutions are produced.

Suppose no article has ϕ as morph (key), normal parsing has failed, and ϕ is an OOV. Then the process restarts in the MOTINC (UnknownWord) mode: the input pointer is at the left of ϕ and the FTM is MODINC, as if the empty string had to be cut as a prefix. That triggers the execution of rule MOTINC, which calls the special function -TRANS- to cut off ϕ and assign ϕ to the UL (as if one had written UL := ϕ).

Examples of morphological analyses by ATEF for tweets are shown in Appendix 7.

```
RDICT : (1,2,3,4/NU)        ** D1 for ending, D2 for radicals(all languages).
                            ** D3 for prefixes (all languages).
                            ** D4 for all word forms.
                            ** D5 for OOV affixes, D6 reserved for future use.

MOTINC: MODINC == CAT:=UNK. ** Simplest form of OOV rule.

RWORD : FTMHIW-FTMGJW-FTMMRW-FTMBNW-FTMXML-FTMEMO: ** Wordform -> result.
        VAR(C):= VAR(A), UL(C) := UL(A). ** Take all information from A, the
       "argument", ie the dictionary entry.
```

Figure 31: Grammar for the ATEF parser

III.1.3.3 **Open components**

We put forth examples of dictionaries of word forms, named entities and idioms in the strict ATEF syntax.

III.1.3.3.1 Dictionaries of word forms

```
'कामनाएं'    == FTMHIW (*,  'कामना').
   FTSEMPTY / CAT-E-N, GEN-E-FEM,
          NUM-E-PLR, PER-E-TD, CAS-E-DIR.

'अरबियों'    == FTMHIW (*,  'अरबी').
   FTSEMPTY / CAT-E-N, GEN-E-MAS, NUM-E-PLR,
          PER-E-TD, CAS-E-OBL.
```

Figure 32: Dictionary of Hindi word forms (strict ATEF syntax)

```
'भक्ताला'    == FTMMRW (*,  'भक्त').
   FTSEMPTY / CAT-E-N, GEN-E-MAS, NUM-E-SNG,
          PER-E-TD, CAS-E-ACC.

'नद्यांनी'    == FTMMRW (*,  'नदी').
   FTSEMPTY / CAT-E-N, GEN-E-FEM, NUM-E-PLR,
          PER-E-TD, CAS-E-INS.
```

Figure 33: Dictionary of Marathi word forms (strict ATEF syntax)

III.1.3.3.2 Dictionaries of « idioms »

```
'बाल-बाल बचा'== FTMHIM (FTSIDIOM,'बाल-बाल_बचा-N_V').
'कुम्भ मेले'  == FTMHIM (FTSIDIOM,'कुम्भ_मेले-N_N').
```

Figure 34: Dictionary of named entities and idioms (strict ATEF syntax)

III.2 Collecting and preprocessing available lexical information in a LexDB

We use LexDB as an abbreviation for 'Lexical Data Base'.

III.2.1 Underlying LexDB (in Jibiki and in Kyoto Cabinet)

III.2.1.1 Rationale

In order to collect the available lexical information between Indian (Hindi, Marathi) languages and English, we identify web resources and programmatically collect the available monolingual and bilingual dictionaries to an extent possible. However, we need a central lexical repository with a unified, structured and consistent interface to be able to integrate and use these resources for SUFT-1. Here, we refer to the JIBIKI-2 platform that makes it possible to process almost all lexical XML resources (Ying, 2016). PIVAX-3 is a lexical database based on JIBIKI-2 that manages the heterogeneous resources by making use of the concept of macrostructures and microstructures.

The macrostructures are represented by metadata describing the types of resource volumes and their relations. The microstructures on the other hand represent the organisation of dictionary entries specific to a volume. PIVAX-3 provides facilities for importing a dictionary or volume, creating and editing entries, and searching the lexical databases by means of a service API (Mangeot-Nagata, 2016).

The pre-requisite to import a dictionary or volume however, is the conversion of inconsistently formatted dictionaries to an XML form. This requires defining a microstructure for the available monolingual or bilingual resource and then transforming or preprocessing the data for import as XML files.

III.2.1.2 Structuring in Jibiki

The structuring of dictionary articles in JIBIKI-2 in principle, follows the Lexical Markup Framework (LMF) standard (Francopoulo et al., 2009): each article contains a form block which includes information related to the form: headword, pronunciation, part-of-speech and a semantic block with a list of sense blocks. Each sense block describes a word meaning. It also contains the translation in another language as well as a list of examples. Each example is translated into the other language. For reasons of convenience, however, the process of structuring articles in JIBIKI-2 has been adapted to gather information about the word form into one "<forme>" block and each word sense into a "<sens>" block.

The step after resource identification is to decide the microstructure specifications and then adapt the resource to those specifications. The approach followed in JIBIKI-2 ensures that the resource complies with the LMF standard while keeping its own tags in the XML format. We import a Hindi UW dictionary containing more than 100K entries into JIBIKI-2 by suitably transforming the raw dictionary data to an appropriate microstructure specification.
An example dictionary entry (raw format) is as follows.

[संगतकार] {} "accompanist(icl>musician)" (N,M,ANIMT,FAUNA,MML,PRSN,TTSM,Na) <H,0,0>;

III.2.1.3 Instantiation

We instantiate among several identified resources, the UW dictionary mentioned in the previous section. The raw data of the dictionary is transformed in an XML format corresponding to a suitable microstructure specification as shown in Figure 35 below.

```
<uc:entry num="545">
   <uc:example>'[संगतकर] {} &quot;accompanist(icl&gt;musician)&quot;
(N,M,ANIMT,FAUNA,MML,PRSN,TTSM,Na) &lt;H,0,0&gt;;'</uc:example>
   <uc:headWord>संगतकर</uc:headWord>
   <uc:ID>{}</uc:ID>
   <uc:UW-Constraint>accompanist(icl&gt;musician)</uc:UW-Constraint>
   <uc:UW-Attributes>N,M,ANIMT,FAUNA,MML,PRSN,TTSM,Na</uc:UW-
Attributes>
   <uc:UW-Flags>
     <uc:language-flag>H</uc:language-flag>
     <uc:frequency-flag>0</uc:frequency-flag>
     <uc:priority-flag>0</uc:priority-flag>
   </uc:UW-Flags>
</uc:entry>
```

Figure 35: Example of a Hindi UW dictionary entry

III.2.2 **Methods to coalesce various resources and import them into our LexDB**

We identified resources (monolingual, bilingual dictionaries, universal word dictionaries, morphological analyzers) differently available for Hindi, Marathi and English, to be later adapted and integrated in PIVAX-3. We present the import of these resources by language.

III.2.2.1 **Hindi**

III.2.2.1.1 Morphological analyzers for Hindi: IITB[65] and IIITH[66] We used two morphological analyzers from which we got analyses of 87049 unique word forms (extracted from a set of collected tweets detailed in III.3.1).

1. A Hindi morphological analyzer available from CFILT, IITB (Chatterjee, Joshi, Khapra, & Bhattacharyya, 2010).

```
---------- Set of Roots and Features ----------
Token : बेटियों,  Total Output : 1
[ Root : बेटी, Class : B, Category : noun, Suffix : यों ]
   [ Gender : -masc, Number : +pl, Person : x, Case : +oblique, Tense : x, Aspect : x, Mood : x ]
   [ Gender : -masc, Number : +pl, Person : x, Case : +oblique, Tense : x, Aspect : x, Mood : x ]
   [ Gender : -masc, Number : +pl, Person : x, Case : +oblique, Tense : x, Aspect : x, Mood : x ]
---------- End of Result ----------
```

Figure 36: Exact output of IITB morphological analysis for Hindi word-form बेटियों (girls)

2. A publicly available Hindi MA from the IIITH website[67].

Address	TOKEN	Features (af='root,cat,gen,num,per,case,tam,suff')
1	बेटियों	<fs af='बेटी, n, f, pl, 3, o, 0, 0'>

Figure 37: Exact output of IIITH morphological analysis for Hindi word-form बेटियों (girls)

Outputs from the analyzers for a more varied set of word forms can be seen in Appendix 4.

[65] IITB = Indian Institute of Technology Bombay, India.
[66] IIITH = Indian Institute of Information Technology Hyderabad, India.
[67] http://sampark.iiit.ac.in/hindimorph/web/restapi.php/indic/morphclient

III.2.2.1.2 Hindi-UW dictionary

We have imported the Hindi UW dictionary[68] which contains 136154 open-class words (64755 lemmas) and 556 closed class words (401 lemmas) available in a plain text format. As a first step, we parsed the dictionary and converted it to XML. An XML entry for this dictionary is shown in Figure 35. The dictionary was integrated in PIVAX-3 and is accessible through the relevant APIs. The dictionary management module of SUFT-1 searches the dictionary online and uses the results for Hindi tweet annotations. Depending on the annotation detail selected by the user, we can either display only the translations, or display a richer structure including the UW constraints as provided by the dictionary.

III.2.2.1.3 Bilingual dictionary: Apertium, Shabdkosh

APERTIUM[69] is an open source platform with numerous associated multilingual resources[70] contributed towards building MT systems. In the present context, we found and used a bilingual 'en-hi' dictionary with 30463 entries. Here is an example of an entry of this dictionary.

```
<e><p><l>twitter<s n="vblex"/></l><r>चहक<s n="vblex"/><s n="iv"/></r></p></e>
```

The metadata of an article include POS level information at various levels of granularity. Similarly, we identified the 'hi-en' from SHABDKOSH[71], which contains several bilingual dictionaries freely downloadable as well as accessible as a web service. The dictionary that we downloaded contained 22756 entries with several translations for each Hindi lemma.

The above two resources have been locally compiled using the KYOTOCABINET hash database for quick access while annotating. We envisage to import it also in PIVAX-3.

III.2.2.2 **Marathi**

For Marathi, we draw from a resource developed and downloadable[72] as a part of a NLP research project done in the CFILT lab. This resource is a Marathi lexicon with 27707 lexical units, each supplemented by a word that represents a morphological paradigm class and a POS tag. The schema is <lexical unit, paradigm class word, POS>. Here is an example.

<संधी>, <मामी>, <noun>

Depending on the morphological paradigm class, we transform the existing data by grouping over each paradigm and then by manually associating the suitable affixes for 'singular/plural' as follows (resulting in more than 50K entries).

```
#Count <pdgmClassWord> ||| <affixes(sing): nom, acc, inst, dat, abl, poss/gen>
 ||| <affixes(plu):nom, acc, inst, dat, abl, poss/gen>
1124  <मामी> ||| *,ला,ने,ला,हून,चा  ||| -या,-यांना,-यांनी,-यांना,-यांहून,ांचा
```

[68] http://www.cfilt.iitb.ac.in/~hdict/webinterface_user/index.php (Total: 136710 UWs, 65156 lemmas)
[69] https://www.apertium.org/index.eng.html
[70] http://wiki.apertium.org/wiki/Documentation
[71] http://www.shabdkosh.com (online bilingual dictionary)
[72] http://www.cfilt.iitb.ac.in/Downloads.html

III.2.2.3 English

For English, we choose to use an existing FST-based morphological analyzer from Open Xerox[73] which is available as a web service for at least 9 more languages.

Several finite-state tools have been employed to build a lexical transducer which is bidirectional (it uses the same finite-state network for analysis and generation), fast and compact.
An example English input tweet with its morphological analysis is shown below.

it's twice's first showcase in japan but it looks like a sold out ⃠ arena concert

```
it
        it +Pron+Pers+NomObl+3P+Sg
's
        's +open+NOUN
twice
        <twice> +Adv
's
        's +open+NOUN
first
        one +Noun+Sg
        one +Num+Ord
showcase
        <showcase> +Verb+Pres+Non3sg
in
        in +Prep
        <in> +Noun+Sg
        <in> +Adj
        <in> +Adv
japan
        japan +open+NOUN
but
        but +Prep
        but +Conj+Sub
        <but> +Noun+Sg
it
        it +Pron+Pers+NomObl+3P+Sg
looks
        <look> +Noun+Pl
        <look> +Verb+Pres+3sg
like
        like +Prep
        like +Conj+Sub
        <like> +Noun+Sg
        <like> +Verb+Pres+Non3sg
        <like> +Adj
        <like> +Adv
a
        a +Let
        a +Det+Indef+Sg
sold
        <sell>ed} +Adj
        <sell> +Verb+PastBoth+123SP
out
        out +Prep
        <out> +Noun+Sg
        <out> +Verb+Pres+Non3sg
        <out> +Adj
        <out> +Adv
⃠
        <⃠> +open+NOUN
arena
        <arena> +Noun+Sg
concert
        <concert> +Noun+Sg
        <concert> +Verb+Pres+Non3sg
```

III.2.3 Work at the level of LexDB

III.2.3.1 Defining the correspondences

As seen in the previous section, resources contain varying degrees of lexical information, so we take steps to make such heterogenous information amenable to integration in a central simplified lexical database, henceforth referred to as LexDB. We decide on a simplified schema for the LexDB as specified in Figure 38 below.

('form', 'lemma', 'stem', 'root', 'affix', 'category', 'gender', 'number', 'person', 'case', 'tense', 'aspect', 'mood', 'POS', 'translationList')

Figure 38: Schema for a central "normalized" lexical database (LexDB)

Depending on the resource and level of information, the data was unified under this schema, to be later adapted and integrated in PIVAX-3 by establishing correspondences between the information units and the schema attributes, and then normalising accordingly.

[73] https://open.xerox.com/Services/fst-nlp-tools/Pages/API%20Docs

We also later transform this unified data to a format expected by the ATEF dictionary compiler after

1. defining a common annotation for morphological analyses of Indian tweets,
2. establishing correspondences between the schema attributes and the annotations.

The two transformations from the dictionaries to the normalized format and from it to the Ariane-G5 format are elaborated in the following subsections.

In the following illustration, we show an example of correspondences established and used during the first conversion from unnormalised resource data into the LexDB format.

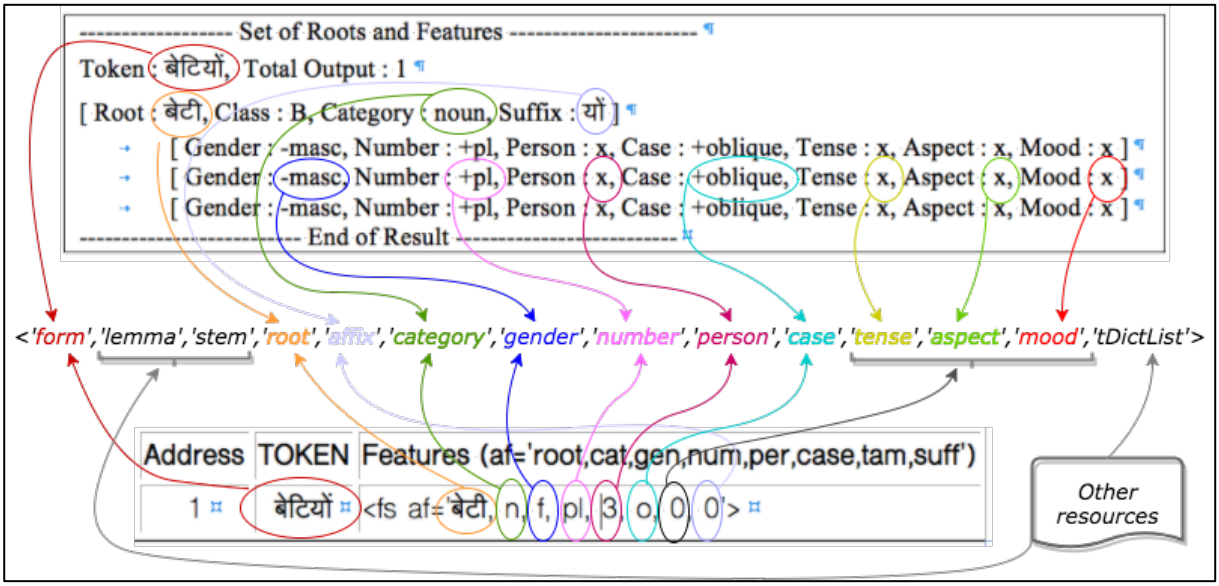

Figure 39: Correspondences and data-flow between output of two morphological analyzers and the LexDB schema attributes

An example of the resulting output is shown below in Figure 40.

<'form','lemma','stem','root','affix','category','gender','number','person','case','tense','aspect','mood','POS','translationList'>													
बेटिया	बेटिया	-	-	-	unk	-	-	-	-	-	-	-	-
बेटियाँ	बेटी	-	-	-	n	f	pl	3	d	-	-	-	-
बेटियाँ	बेटी	-	-	याँ	noun	-masc	+pl	x	-oblique	x	x	x	-
बेटियां	बेटी	-	-	-	n	f	pl	3	d	-	-	-	-
बेटियां	बेटी	-	-	यां	noun	-masc	+pl	x	-oblique	x	x	x	-
बेटियो	बेटियो	-	-	-	unk	-	-	-	-	-	-	-	-
बेटियों	बेटी	-	-	-	n	f	pl	3	o	-	-	-	-
बेटियों	बेटी	-	-	यों	noun	-masc	+pl	x	+oblique	x	x	x	-
बेटियोंसे	बेटियोंसे	-	-	-	unk	-	-	-	-	-	-	-	-
बेटी	बेटी	-	-	-	n	f	sg	3	d	-	-	-	-
बेटी	बेटी	-	-	-	n	f	sg	3	o	-	-	-	-

Figure 40: Example of a populated LexDB table

However, before establishing correspondences from the unified data to Ariane-G5 format, we need to define common notations to be utilised later for ATEF.

III.2.3.2 **Defining the common notation used in the morphological analyzer**

We define "syntactic" (actually syntactic, morphological or semantic) variables (attributes) for an MA phase aiming at handling Indian tweets, potentially multilingual: containing Hindi, Marathi and English.

For this, we refer to (Sankaran et al., 2008) that works out a framework for tagsets including morphosyntactic features covering most Indian languages. We borrow from the hierarchical schema reported in the paper and adapt it to build a common annotation resource for ATEF-based MA (DVS).

We declare the morpho-syntactic variables under exclusive and non-exclusive categories. Additional variables suitable for incorporating lexical units from Indian tweets have also been introduced.

```
** Subcategory of emotional signs.
SUBEM == (EMOLIST, ** EMoji list.
          PHATIC   ** Hmmm!, Uh!, Aha! etc.
         ).
** Subcategory of tweet-specific occurences.
SUBTW == (TWCD,  ** TWeet COmmand (such as RT for ReTweet).
          TWAD   ** TWeet ADdress (such as @Ritesh).
          ).
```

Figure 41: Example of variable declarations suited for tweets belonging to the exclusive sub-categories

```
-NEX- ** non-exclusive variables:
         a value is any subset of the list of elementary values.

** Morphosyntactic category (for terminals in a classical PSG).
CAT  == ( N,    ** Noun.
          V,    ** Verb.
          P,    ** Pronoun.
          J,    ** Nominal modifier (adJective).
          D,    ** Demonstrative.
          A,    ** Adverb.
          L,    ** participLe.
          PP,   ** PostPosition.
          C,    ** partiCle.
          PU,   ** PUnctuation.
          EM,   ** EMoji or phatic.
          OT,   ** Out of Text (hors-texte in French).
          TWCD,** TWeet COmmand (such as RT for ReTweet).
          RD    ** ResiDual.
        ).
** Morphological Mood of inflected verbs.
MOOD  == (DCL,    ** DeCLarative.
          SBJ,    ** SuBJunctive.
          CND,    ** CoNDitional.
          IMP,    ** IMPerative.
          PSM,    ** PreSuMptive.
          ABT     ** ABiliTative.
         ).
```

Figure 42 : Example of variable declarations belonging to the non-exclusive categories

Appendix 3 contains the complete variable declarations.

III.2.3.3 **Normalization of attributes and values in the LexDB**

Using the common annotation developed in the previous section and the data unified under the schema, we build a resource conforming to Ariane-G5 format. We see in Figure 43 that data unified under the schema have different values, depending on the different resources they come from.

Here are some examples of attributes and values from the LexDB, and the correspondences used for conversion. A detailed table is provided in Appendix 5.

LexDB values	Transformed canonical form	LexDB values	Transformed canonical form
(schema attribute: root)		*(schema attribute: tense)*	
'-', 'x'	CAT(CAT0)	'pa', '+past', '(+past'	TNS(PST)
'n', 'noun'	CAT(N)	*(schema attribute: person)*	
'proper noun'	CAT(N), SUBN(P)	'2h', '(1p:2phon:3p)	PER(SDHN)
'v', 'verb'	CAT(V), SUBV(M)	*(schema attribute: gender)*	
'verb_aux'	CAT(V), SUBV(A)	'any', '+-masc'	GEN(MAS, FEM)
'sh_n', 'psp', 'post position'	CAT(PP)	*(schema attribute: number)*	
'adj', 'adjective'	CAT(J), SUBJ(ADJ)	'-', 'x'	NUM(NUM0)
'quantifier	CAT(J), SUBJ(Q)	'sg', '-pl'	NUM(SNG)
'interrogative pronoun', 'interrogative'	CAT(P), SUBP(WH)	*(schema attribute: case)*	
'interjection'	CAT(EM)	'o', '+obl', '+oblique'	CAS(OBL)
'unk'	CAT(RD)	*(schema attribute: mood)*	
		'subjunctive','+conditional+subjunctive'	MOOD(SBJ)
		'+conditional','+imperative+conditional', '+ability+conditional', '+conditional+subjunctive'	MOOD(CND)

Figure 43: Multiple values under different attributes of the LexDB converted to a canonical form

Figure 43 shows the canonical forms obtained from the conversion, and these forms are used to construct an entry in the Ariane-G5 format for each lexical unit.
A few such entries which are constructed and passed to ATEF to build dictionary entries are shown in Figure 44 below.

'बेटिया': UL('बेटिया'), AFFIX(AFFIX0), CAT(RD), GEN(GEN0), NUM(NUM0), PER(PER0), CAS(CAS0), TNS(TNS0), ASP(ASP0), MOOD(MOOD0).

'बेटियाँ': UL('बेटी'), AFFIX(AFFIX0), CAT(N), GEN(FEM), NUM(PLR), PER(TD), CAS(DIR), TNS(TNS0), ASP(ASP0), MOOD(MOOD0).

'बेटियाँ': UL('बेटी'), AFFIX('याँ'), CAT(N), GEN(FEM), NUM(PLR), PER(PER0), CAS(DIR), TNS(TNS0), ASP(ASP0), MOOD(MOOD0).

'बेटियां': UL('बेटी'), AFFIX(AFFIX0), CAT(N), GEN(FEM), NUM(PLR), PER(TD), CAS(DIR), TNS(TNS0), ASP(ASP0), MOOD(MOOD0).

'बेटियां': UL('बेटी'), AFFIX('यां'), CAT(N), GEN(FEM), NUM(PLR), PER(PER0), CAS(DIR), TNS(TNS0), ASP(ASP0), MOOD(MOOD0).

'बेटियो': UL('बेटियो'), AFFIX(AFFIX0), CAT(RD), GEN(GEN0), NUM(NUM0), PER(PER0), CAS(CAS0), TNS(TNS0), ASP(ASP0), MOOD(MOOD0).

'बेटियों': UL('बेटी'), AFFIX(AFFIX0), CAT(N), GEN(FEM), NUM(PLR), PER(TD), CAS(OBL), TNS(TNS0), ASP(ASP0), MOOD(MOOD0).

'बेटियों': UL('बेटी'), AFFIX('यों'), CAT(N), GEN(FEM), NUM(PLR), PER(PER0), CAS(OBL), TNS(TNS0), ASP(ASP0), MOOD(MOOD0).

'बेटियोंसे': UL('बेटियोंसे'), AFFIX(AFFIX0), CAT(RD), GEN(GEN0), NUM(NUM0), PER(PER0), CAS(CAS0), TNS(TNS0), ASP(ASP0), MOOD(MOOD0).

'बेटी': UL('बेटी'), AFFIX(AFFIX0), CAT(N), GEN(FEM), NUM(SNG), PER(TD), CAS(DIR), TNS(TNS0), ASP(ASP0), MOOD(MOOD0).

'बेटी': UL('बेटी'), AFFIX(AFFIX0), CAT(N), GEN(FEM), NUM(SNG), PER(TD), CAS(OBL), TNS(TNS0), ASP(ASP0), MOOD(MOOD0).

Figure 44: Entries suitable for the ATEF component in the Ariane-G5 format

III.3 Generation of ATEF dictionaries from the LexDB

III.3.1 Hindi

As seen in the previous section, we use two morphological analyzers to process 87K distinct word forms from Hindi tweets. We obtain the 87K word forms by extracting Hindi tweets using the TWITTER search API with the query "lang:hi" and then by applying some text processing. Text processing is necesssary to obtain only Devanagari scripted words (so they can be processed by the Hindi MA) .

The text messages within the tweet JSON data structure contain **hashtags** (words prefixed by the '#' symbol), **usernames** (words prefixed by the '@' symbol), **web links** (URLs) and '**RT**' (when placed at the beginning of a message indicates the action of 'retweet' to the message, this has remained so since TWITTER began and has been retained for backward compatibility). In addition, Hindi tweets acquired in spite of using the 'lang:hi' advanced operator contain many romanised words (belonging to ASCII charset). Therefore, text processing also helps to filter out the aforementioned TWITTER specific terms, the web links and the romanised word forms.

We acquire a total of 245394 unique tweets with a vocabulary of 520047. After filtering out the TWITTER specific terms and web links, the vocabulary reduces to 352150. These word forms are processed and separated into three mutually exclusive groups.

III.3.1.1 158371 words consisting of only ASCII characters, & punctuations

||| gai ||| SP.. ||| team!:) ||| godaan ||| gad ||| booty ||| murekhh ||| bhadwo, ||| Euro ||| aigooo ||| Valle ||| chachipoyaru ||| how: ||| wooden ||| lehron ||| Sach ||| Sack ||| patekarancha ||| pahchaniye ||| Isabella ||| dayum, ||| "Carrabelles"? ||| jhee ||| jhel ||| ....bangkok ||| >#Pakistan ||| boleros. ||| Dog2:Tu ||| yogen198: ||| Mucky ||| Happening ||| caner ||| gaya..?? ||| sy..!! ||| lagwao ||| chunusi ||| rebel ||| zoology... ||| zadarchee, ||| OLLY ||| Abbu ||| Restorate ||| 160253 ||| dna ||| improvisation ||| want... ||| Thukam ||| ekuruvanee ||| 1JUV ||| yahoo ||| Indigo ||| Sulajh ||| 2015" ||| Ago ||| A.7) ||| ShivshankarS: ||| 10hz... ||| mieux ||| Editorials ||| krain ||| Zikr ||| congress,jinhone ||| URUGUAAY ||| Unavijika ||| balada ||| Phoppu ||| Alahaiii ||| shreya ||| naw? ||| CARTOON ||| Sun-Wolf ||| beparwah ||| Valobashe ||| welcomes ||| fir ||| fit ||| fix ||| "@Shree_15: ||| Alderley ||| fii ||| fin ||| fio ||| fil ||| fim ||| shivay ||| welcome! ||| welcome" ||| RAGGAE: ||| welcome:

Figure 45: Word forms from Hindi tweets consisting of ASCII characters and punctuations (' ||| '~delimiter)

III.3.1.2 174734 words consisting of only non-ASCII characters (including emojis) and punctuations

||| सुनाके ||| घाटा ||| आमलकी ||| संज़ोया ||| पड़े ||| पड़ी ||| सौंपा: ||| जाएं.... ||| इत्र ||| सौंपा, ||| खांसता ||| करंडक ||| पड़ल ||| पड़ा ||| मे...💞💞 ||| काळात... ||| गए?😂😏👏 ||| मास्साब ||| घाटी ||| घाटे ||| सुनाकर ||| घाटो ||| 🐯👌🐯👌▶अत्यंत ||| डी.सी. ||| ✅ ||| जी भारत ||| है..⚡⚡ ||| दश्त-ए-फ़लक ||| संस्थांविरोधात ||| [それでも君が好きだよ] ||| पिट्रोल ||| नारेबाजी ||| खरचो ||| गाएँगे ||| अंग्रेजी, ||| होगा..सब ||| जल💥 ||| विधायको ||| हो...😜😈😆😂 ||| ?कहां ||| व्यास ||| व्याह ||| व्यार ||| व्याप ||| सुकमा-कोंटा ||| मिल्ली..पानी ||| तो.सारे ||| विधायक। ||| बन्दों! ||| त्योहार!... ||| सुलझवा ||| नूपुर ||| चाँद🤔 ||| पड़ो ||| चाँद🌙 ||| लड़ेंगे ||| हूंगी ||| उडीसा ||| कातरवेळी... ||| गुमसुम ||| गडावर ||| सोईनुसार ||| है।#कमीने ||| दिया!मतलब ||| ध्वनिमत ||| जायेगा"--आज़म ||| बनवातीं.. ||| दें,कार्यकर्ता ||| मोदी-दीदी ||| खांसते ||| काढ़ुन ||| मलतब ||| लहर... ||| お願いします🙇 ||| (कैलारस,मुरैना)पर ||| नही।यह ||| लहर" ||| "पाकिस्तानियों ||| मर्दों ||| प्रोफ़ाइल ||| 😛😛😅😅 ||| विधिवत

Figure 46: Word forms from Hindi tweets consisting of non-ASCII characters, emojis and punctuations ('||| '~delimiter)

III.3.1.3 **19045 words containing both ASCII and non-ASCII characters**

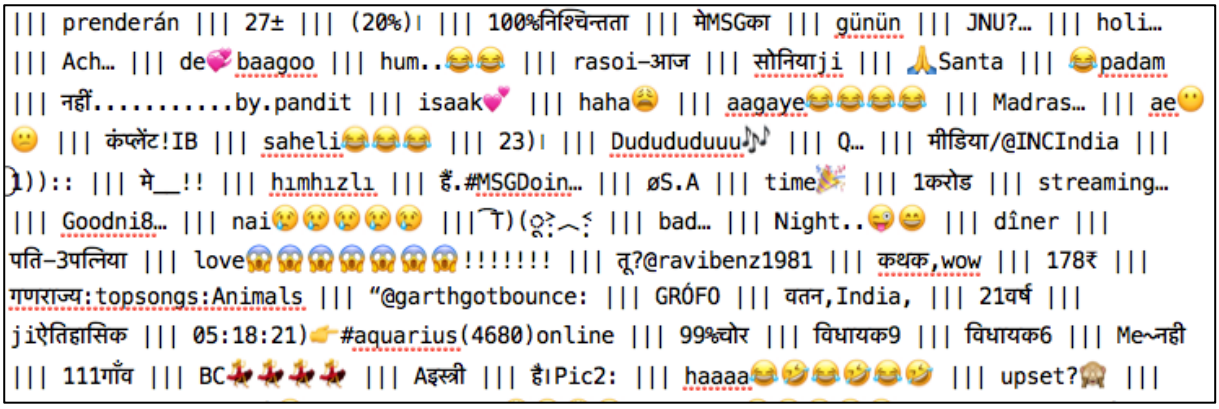

Figure 47: Word forms from Hindi tweets consisting of both ASCII, non-ASCII characters and punctuations (' ||| '~delimiter)

We further process the 174734 word forms from the second group, separate out punctuations and emojis to get a final 87049 word forms of purely Devanagari scripted Hindi word forms which we use for MA.

III.3.2 **Marathi**

Using the TWEEZER program, we extract word forms from Marathi tweets in a similar way as done for Hindi word forms in the previous section. The only difference is that, in acquiring the Marathi tweets, we give the query operator 'lang:mr' to the TWITTER API instead of 'lang:hi'.

We acquire a total of 85685 unique tweets with a vocabulary of 226220. After filtering out the TWITTER specific terms and web links, the vocabulary reduces to 153767. These word forms are processed and separated into three mutually exclusive groups.

III.3.2.1 **7204 words consisting of only ASCII characters, & punctuations**

Figure 48: Word forms from Marathi tweets consisting of ASCII characters and punctuations (' ||| '~delimiter)

III.3.2.2 143387 words consisting of only non-ASCII characters (includes emojis) and punctuations

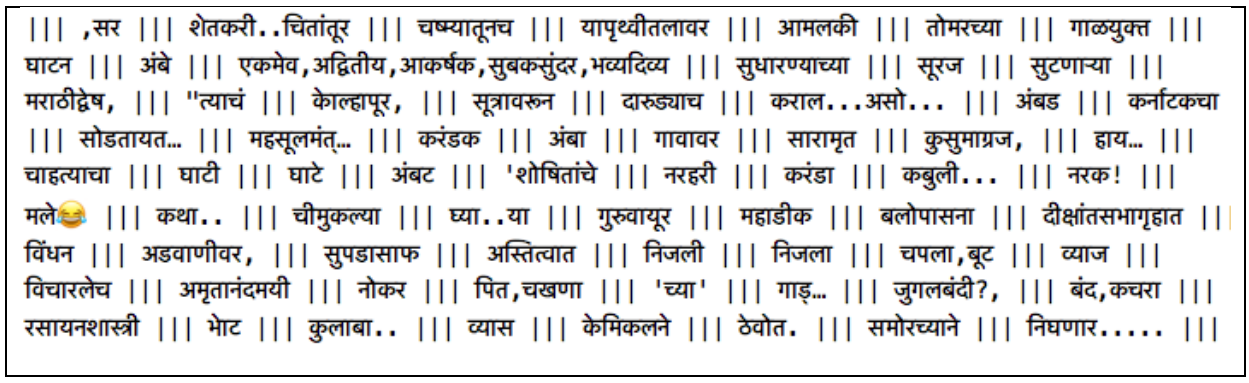

Figure 49: Word forms from Marathi tweets consisting of non-ASCII characters, emojis and punctuations (' ||| ' ~ delimiter)

III.3.2.3 3176 words containing both ASCII and non-ASCII characters

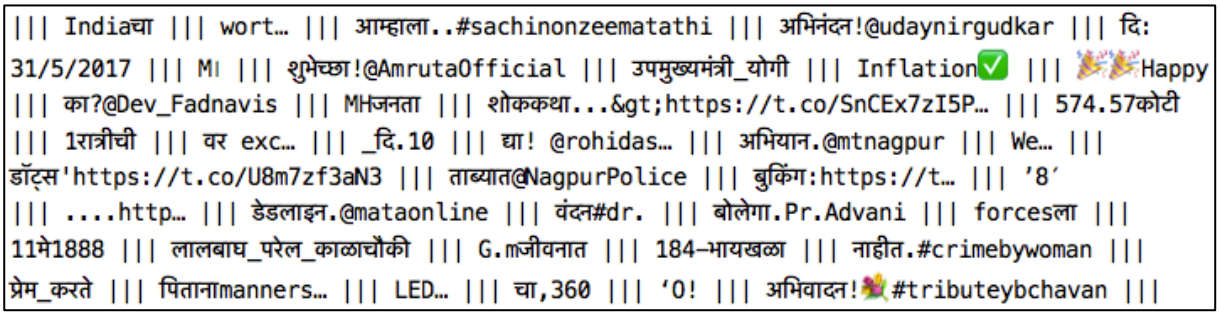

Figure 50: Word forms from Marathi tweets consisting of both ASCII, non-ASCII characters and punctuations (' ||| ' ~ delimiter)

We further process the 143387 word forms from the second group, separate out punctuations and emojis to get a final 95877 word forms of purely Devanagari scripted Marathi word forms which we use for MA. If we ignore the third group of word forms containing both ASCII and non-ASCII (~2%), the proportion of only ASCII word forms is about 4.8%.

In Table 24, we provided a more detailed account of the tweets collected using the TWEEZER[74] program which allows us to increase the vocabulary of tweets collected over a period of time, both for Hindi and Marathi.

III.3.3 English

As seen previously for Hindi and Marathi tweet word forms extraction, we used the Tweezer program with the query operator "lang:en" for obtaining English tweets and acquired a total of 91561 unique tweets with a vocabulary of 255599 word forms. After filtering out the TWITTER specific terms and web links, the vocabulary size reduces to 129257. These word forms are processed and separated into three mutually exclusive groups (only ASCII characters, only NON-ASCII characters and "ASCII+NON-ASCII" characters).

The total number of word forms with only ASCII characters from the filtered vocabulary of 129257 is 115703 (89.5%) with the remaining two groups standing at 3.1% and 7.4% respectively.

Figure 51 shows a few examples from the first group.

[74] https://bitbucket.org/riteshms/tweezer

||| Fame. ||| EXPLAIN ||| sowell ||| chameleons ||| artifacts. ||| spiders ||| Aamiin.... ||| gab ||| virtuoso, ||| everything.i ||| Retreat ||| Euro ||| Vitamine ||| Tut ||| pages ||| wood, ||| WATERMELON ||| RADIO, ||| stodgy, ||| bringing ||| wooden ||| bitch!!! ||| $UNG ||| wednesday ||| everything." ||| Reidy's ||| everything.. ||| 'Pacific ||| 27" ||| Dansak ||| noticing.. ||| TRAI ||| WOW!! ||| 27, ||| Shocked ||| NaVi. ||| 270 ||| quagmire. ||| affiliate. ||| 275 ||| 276 ||| 278 ||| 279 ||| defenses ||| Pharmacologist ||| Residency. ||| warmongering ||| INDIAN'S ||| replaced ||| kid, ||| kid. ||| TRAP ||| Happening ||| Weeks: ||| WENZY ||| fire, ||| affiliated ||| Manger ||| kids ||| uplifting ||| MATCHES ||| session... ||| 27P ||| Off... ||| MATCHED ||| OLLI ||| [TEASER] |||

Figure 51: Word forms from English tweets consisting of ASCII characters and punctuations (' ||| '~delimiter)

III.4 Experimentation and evaluation

III.4.1 Settings

III.4.1.1 Small quantity vs. large quantity

For evaluation, we propose to have an experimental setting with a small quantity of tweets(100), and another with a large quantity of tweets (250K) as shown in Table 13.

III.4.1.2 User-controlled vs. spontaneous

Another division of tweets is based on whether they are user-controlled or spontaneous, which was the case. To collect 'user-controlled' tweets, we used official (e.g. news, government etc.) TWITTER streams, expecting a large proportion of well-formed formal tweets.

We collected 'spontaneous' tweets directly by starting from users and extending to followers. The tweets from this set are more disfluent and informal.

Table 23: User-controlled and spontaneous tweet sets

Type of tweets	TWITTER handles	Quantity
User-controlled	'@bharatkhabarweb', '@divyabhaskar', '@zeenewshindi.	10K
Spontaneous	'@nanditathakur', '@ac_sk8298149', '@SaffronRocks'	10K

III.4.2 Evaluation of quality

III.4.2.1 Methodology

For evaluating the quality of the MA resources, we examined a small sample (for Hindi and Marathi).

We have in fact a large gold standard (~ 220K entries). Because of the method used, its quality must be the same as that of our resource. However, if we had extended this MA by adding subgrammars to handle the decomposition in [prefix] radical [suffix], using a dictionary of affixes, and a dictionary of radicals, quality (as inverse of distance between the answers and the corrected answers) would not be 100% at the beginning of the deployment.

III.4.2.2 Quality of the resource

Concerning tweets, what is their "quality"? We will consider that the resource we used is "perfect by construction", because we use only genuine tweets and recover them with all their properties and contents.

Regarding the lexical resources for Indian languages, no expert evaluation has been done so far. Although we have no real competence in morphology and grammar, we have examined a random set of 100 items in each of our resources (for Hindi and for Marathi), and found very few errors (1-2%) in the monolingal resources, but again we are not really qualified to do that kind of evaluation.

For the HI-UNL dictionary, we found less than 0.5% erroneous entries.

Here is an example of an erroneous entry from the HI-UNL dictionary.

[वेवेकअ] {} "reason(icl>discretion)" (N,M,INANI,Na) <H,0,0>;

III.4.2.3 **Quality of the output of morphological analyzer**

For Hindi and Marathi, the quality of the output of the morphological analyzer is the same as that of the resource used, because we simply perform a table retrieval. We could improve it if we had used the ATEF facility for handling compound words, but we did not have enough time to do it.

III.4.3 **Evaluation of coverage of morphological analyzer**

For estimating the coverage of the Hindi morphological analyzer built using the 87049 tweet word forms (cf. III.2.2.1.1), we used the same set of word forms and calculated the coverage against the morphological analyzer resource constructed. The number of corresponding lemmas was 68788 and the coverage obtained was 79.2%[75].

III.4.4 **Evaluation of end-to-end coverage**

To get the end-to-end coverage, we have to multiply by the proportion of lemmas found (as headwords) in our HI-UNL dictionary. It is 13.9%, as only 9041 of the above 68788 lemmas are contained in the HI-UNL dictionary with 65156 lemmas as shown in Figure 52 below.

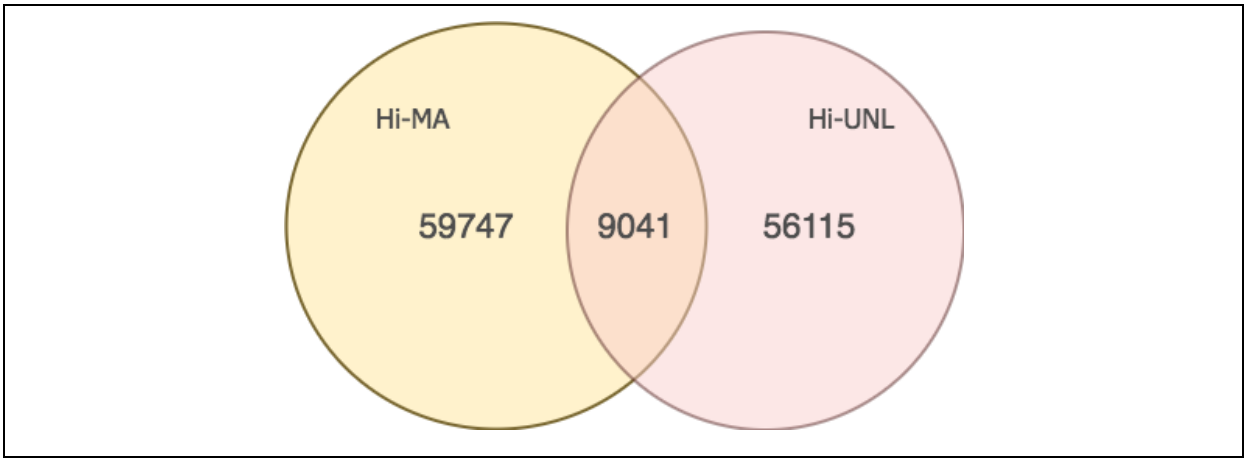

Figure 52: Proportion of lemmas found in Hi-UNL dictionary

All that taken together leads to an end-to-end coverage of 11.06%.

Considering the coverage in terms of meanings and hence of translation possibilities, we are not in a position to propose an evaluation, because no evaluation of the coverage of the HI-UNL dictionary in terms of word senses has been done so far.

[75] We use the simple formula: Coverage = |W|-|OOV|)/|W|.

Synthesis

In this chapter, we first saw that there are no morphological analyzers suited to handling tweet-like texts. That motivated us to build a large-scale multilingual analyzer, which would be specifically suited to code-mixed Indian tweets and could be used by SUFT-1 for achieving better coverage for its AR annotations on tweets.

We designed the ATEF lingware components to be able to handle Indian language tweet word forms *together* with the same MA. We described the methods in detail for constructing a coherent lexical database, which was then used for generating the ATEF dictionaries.

Towards the end, we made evaluations about the quality and coverage, and observed that

1. the quality of the MA resources is good enough, with only 1-2% errors.
2. the quality of the MA output is as good or bad as the resource.
3. the coverage of the MA resource is about 79%, but the end-to-end coverage is much too low at 11%.

Chapter IV Experimentation and evaluation of SUFT-1

Introduction

In Chapter IV, we review our work by presenting the resources integrated in SUFT-1, the description of the actual system and an account of evaluations performed in various language and interface settings

We present an account of the evaluation experiments done for Japanese and two Indian language tweets. We conclude by commenting on the results obtained from end-to-end experiments in 3 settings, namely “hi-en”, “jp-fr”, “jp-en”.

IV.1 Integration of bilingual resources

IV.1.1 Method

We acquired bilingual dictionaries for hi-en and mr-en language pairs and integrated them directly to be accessed by SUFT-1. We were also able to acquire, integrate and access a UNL based UW Hindi dictionary. We elaborate on these resource integrations in the following section.

IV.1.2 Direct integration from bilingual resources

IV.1.2.1 From Hindi-English resources

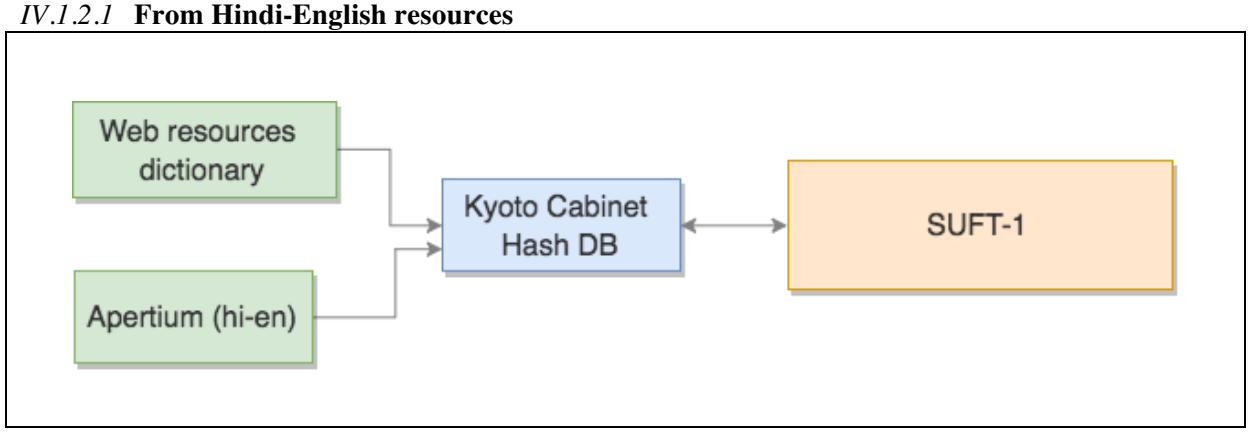

Figure 53: Direct integration of hi-en bilingual dictionaries

A “hi-en” bilingual dictionary of size 58K denoted as ‘Web resources dictionary’ in Figure 53 was created by programmatically collecting data from various web resources. Another open-source dictionary from APERTIUM contributed 31K dictionary entries. Both these dictionaries were coalesced and then transformed in a ‘.TSV’ format. They were then compiled into a Hash DB (.KCH files) using the KYOTOCABINET program.

After compilation, the ‘Web resources dictionary’ increased in size from 870KB to 7.4MB and the APERTIUM dictionary DB size increased to 7.5MB from 962KB. The dictionaries were directly integrated in SUFT-1.

IV.1.2.2 From Marathi-English resources

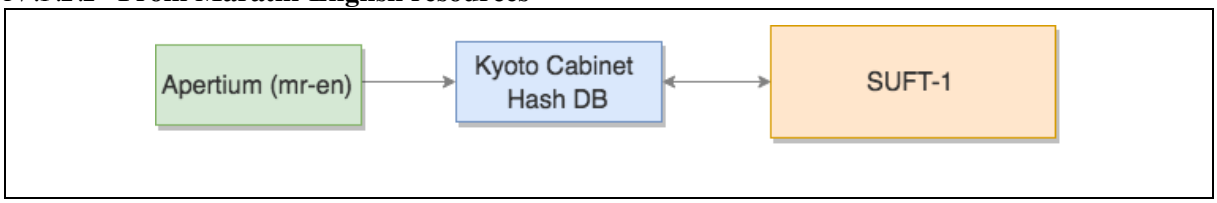

Figure 54: Direct integration of mr-en bilingual dictionaries

We obtained a "mr-en" bilingual dictionary from APERTIUM with 5986 word forms (201K in size). This was processed and, compiled in a KC Hash DB (6.3MB in size) and then integrated in SUFT-1 as shown in Figure 54.

IV.1.2.3 From Japanese-French and Japanese-English resources

We used the online CESSELIN API[76] to access the CESSELIN "jp-fr" dictionary with 82K[77] entries and the online JMDICT API[78] to access the JMDICT "jp-en" dictionary with 48K[79] entries. We make use of XMLHTTPREQUEST APIs to fetch dictionary translations from these resources as shown in Figure 55 below.

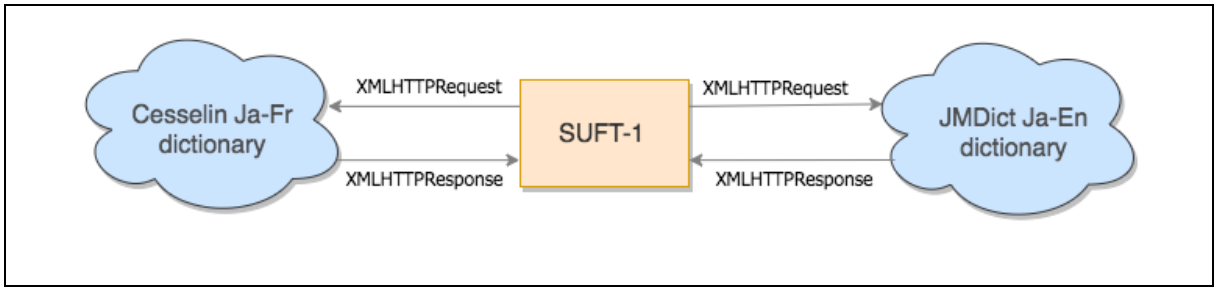

Figure 55: Direct integration of online Jibiki resources for jp-fr and jp-en

IV.1.3 Indirect integration through intermediate language of UNL-based resources

IV.1.3.1 Hindi-UNL-English

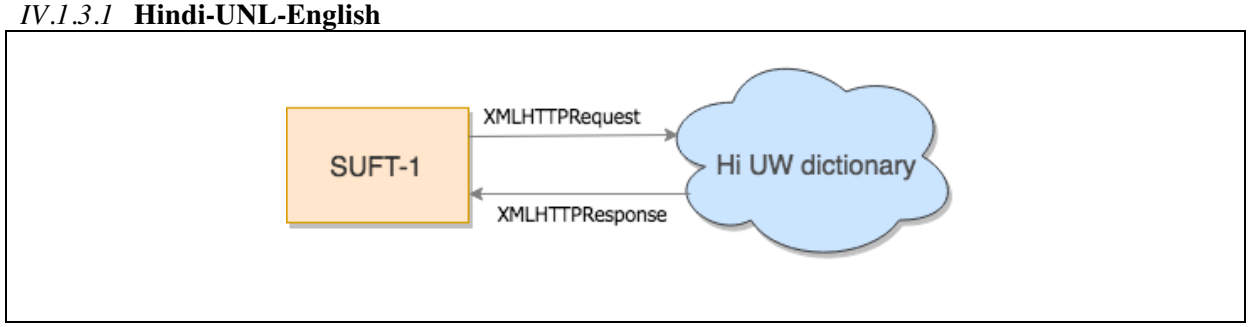

Figure 56: Indirect integration of Hindi Universal dictionary for hi-en annotation

We acquired the Hindi Universal Word dictionary with around 136K universal words (65158 unique Hindi word forms) from the IITB CFILT website.

We transformed this Hindi UW dictionary and uploaded it to the JIBIKI/PAPILLON platform to be accessible through XMLHTTPREQUESTs of the API.

[76] http://jibiki.fr/jibiki/api/Cesselin/jpn/cdm-headword/%E5%B1%B1/cdm-translation?strategy=EQUAL
[77] http://jibiki.fr/statistiques.php
[78] http://papillon.imag.fr/papillon/api/JMdict/jpn/cdm-headword/%E3%81%92/cdm-translation?strategy=EQUAL
[79] http://jibiki.fr/statistiques.php

IV.2 Description of the actual system

IV.2.1 Software configuration

IV.2.1.1 Possibilities for users

IV.2.1.1.1 *Actual screen*

Figure 57: User screen with annotations (jp-en)

Figure 58: Screen with multiple layouts for user facilitation

IV.2.1.1.2 Available features

For the user, SUFT-1 allows :

1. annotating tweets from hi->en, jp->en and jp-fr.
2. importing tweets for annotation from plain text files.
3. searching for tweets and annotating them
4. navigating of tweets one at a time (includes keyboard shortcuts 'right/left arrow keys'
5. selecting of appropriate dictionaries
6. annotating with 1 offline dictionary and 1 online UW dictionary for hi -> en
7. annotating with 2 online dictionaries (CESSELIN and JMDICT) for jp->fr and jp->en respectively
8. viewing optional MT output (YANDEX for now)
9. viewing annotations in 3 layouts for now (as dropdown (2 variations) and tooltip mode)

IV.2.1.1.3 Not yet available features

1. The system is not yet able to store user preferences (for example, dictionary choices, most often used language pair, etc.). It also cannot yet store the user's lexical selections and recall them for subsequent sessions.
2. Integration of MT output from Google Translate remains to be implemented.
3. Annotations accompanied by visualisation of the associated compatibility graph is also not yet implemented.

IV.2.1.2 **Possibilities for evaluators**

IV.2.1.2.1 Actual screen

Figure 59: Screen with annotations and evaluation controls (jp-fr)

IV.2.1.2.2 Available features

For an evaluator, in addition to the user features mentioned previously SUFT allows:

1. evaluating tweets as understandable or not using button controls at the bottom.
2. automatic storing of evaluation time, understandability and other observables in a SQLITE DB.
3. exporting logs for an evaluation session.

IV.2.1.2.3 Not yet available features

A feature that could be added for the evaluators is a DB view within the system so that the evaluator could select sessions or evaluate interesting results without quitting the system or without having to refer to a log for quick analyses.

IV.2.1.3 **Possibilities for developers**

SUFT-1 is built so that it allows a configurable layout (using UIKIT + JS), keeping portability in mind. However, it also uses PHP, which could be replaced with an appropriate technology (CORDOVA) to make this system available across multiple devices, including smartphones and tablets.

The developers can integrate other MT APIs in SUFT-1 for facilitation of the user or more importantly for an evaluator so that s/he can compare the aid provided by various active reading displays and by different MT systems. Alternative MA also can be integrated. The system will be publicly available on BITBUCKET repository.

IV.2.1.4 **Generic and specific aspects**

IV.2.1.4.1 Generic aspects

Lemma-based requests can be performed using LEXTOH or some alternative APIs, which allow such capability.

IV.2.1.4.2 Specific aspects

The use of KYOTOCABINET Hash DB for implementing access to local dictionaries requires installation of KC DB and preparation of the dictionaries.

IV.2.1.4.3 Portability aspects

SUFT-1 presently uses the PHP server-side solutions. It is now usable as a web-service or can be adapted using solutions like CORDOVA for deployment on mobile devices.

IV.2.2 Lingware configuration

IV.2.2.1 **Available resources for tweets in Indian languages**

IV.2.2.1.1 Tweet resources

For tweets, we use TWEEZER program to (cf. III.3 for analyses done using this program).

In Table 24 below, we give a summary of all the tweets collected during the PhD for Hindi and Marathi.

Table 24: Summary: 267417 Hindi and 115619 Marathi tweets were extracted

	Unique tweets / Total tweets retrieved	Vocabulary size and number of different types of terms in the vocabulary							Code-mixing %
Language		#Vocabulary	#Hash tags	#User names	#Filtered	#ASCII	#Non-ASCII	#Mixed code	#ASCII *100/ #Filtered
Hindi	267417/ 462110	562202	30285	75083	377385	168961	187402	21022	44.7
Marathi	115619/ 239878	291795	15188	21089	195238	14889	175851	4498	7.6

The user can also access tweet streams on the fly using SUFT-1 tweet request controls (in the top frame).

IV.2.2.1.2 Monolingual lexical resources

We constructed a large resource of "word forms" appearing in Indian tweets with their morphological analyses for Hindi and Marathi.

1. Hindi: 163221 Hindi word forms from 68788 lemmas.
2. Marathi: 72312 Marathi word forms from 6026 lemmas.

IV.2.2.1.3 Bilingual lexical resources

We obtained the as a bilingual resource

1. HI-UNL dictionary from CFILT-IITB with 136710 Universal words and 65156 lemmas.

2. Hindi-English Shabdkosh dictionary with 22756 entries.

3. Hindi-English Apertium dictionary with 30463 entries.

IV.2.2.2 **Available resources for tweets in Japanese**

IV.2.2.2.1 Tweet resources

For Japanese tweets, we use a collection of 3.2M tweets collected by Prof. Kitamoto (NII, Tokyo) during the whole of February 2014, using 'snow' as query word.

We used samples from this collection to make evaluation experiments on Japanese tweets within SUFT-1.

IV.2.2.2.2 Lexical resources for Japanese-French and Japanese-English

Resources available on the JIBIKI/PAPILLON platform[80] have been used for SUFT-1:

1. Japanese-French: The CESSELIN dictionary, with 82K entries.
2. Japanese-English: The JMDICT dictionary, with 48K entries.

[80] http://jibiki.fr/statistiques.php

IV.3 Methodology of evaluation

IV.3.1 Evaluation of experiments with Indian language tweets

In order to evaluate the help provided by SUFT-1 for understanding Indian language tweets, we performed a closed test-sets experiment with SUFT-1 to evaluate 100 Hindi tweets ('hi-en-misc-100' cf. Table 13).

We carried out experiments for the following view settings:

- "MT output only (MTO)",
- "annotations only (ARO)" and
- "annotations + MT output (AMO)".

We used the "hi-en" language pair setting (hence, with annotations in English) of SUFT-1 for this experiment, and requested one native English speaker (not knowing Hindi) to evaluate the understandability of Hindi tweets annotated with English translations in the horizontal word-by-word layout.

In all settings, we asked each participant to label a tweet as "understandable", *"if the tweet made some sense to her"* and "non-understandable" otherwise. During the evaluations, the *understandability decision* and *understandability decision time* were recorded and logged for each tweet by SUFT-1.

The results of these evaluations for each of the view settings mentioned above were averaged and tabulated for further analyses.

IV.3.2 Evaluation of experiments with Japanese tweets

For experiments on understandability of Japanese tweets, we used the same three view settings (ARO, MTO and AMO) as above. We used one test set[81] of 100 Japanese tweets ('jp-en-snow-100' cf. Table 13) and performed evaluations on them for the "jp-en" (showing English annotations) and "jp-fr" (showing French annotations) language settings.

We requested one English speaking participant and one French speaking participant (both not knowing Japanese) to evaluate the tweets using SUFT-1 with the same evaluation methodology for rating understandability, as described in the previous section.

We averaged the scores logged by SUFT-1 and tabulated them for analyses.

IV.4 End-to-end experiments in 3 settings

IV.4.1 Indian languages-English

Table 25: Evaluation results for experiments on "hi-en" tweets done with AR

Hindi UW (hi-en) dictionary			
Setting / Measures	(Yandex MT) output only	annotations only	annotations + (Yandex MT) output.
Avg. Understandability ratio (%)	15	20	26
Avg. Understandability decision time (sec)	14	70	140

[81] This test set is used for both 'jp-en' and 'jp-fr' settings because we use the same tweets in SL (Japanese).

From the evaluation results in Table 25, we see that the understandability ratio increased by about 5% when replacing the MT output by the AR annotations. It further increased by 6% in the MT+AR setting.

We also note that the average understandability decision time doubles if MT is visible with the AR.

IV.4.2 **Japanese-French**

Table 26: Evaluation results for experiments on "jp-fr" tweets done with AR

Cesselin (jp-fr) dictionary			
Setting / Measures	(Yandex MT) output only	annotations only	annotations + (Yandex MT) output.
Avg. Understandability ratio (%)	12	6	8
Avg. Understandability decision time (sec)	18	20	25

From the evaluation results in Table 26, we see that the average understandability ratio decreased in the MT+AR setting and was even less than when viewed with MT alone, which is not possible.

Hence, we discarded these results. We plan to redo the same experiments as soon as possible.

IV.4.3 **Japanese-English**

Table 27: Evaluation results for experiments on "jp-en" tweets done with AR

JMDict (jp-en) dictionary			
Setting / Measures	(Yandex MT) output only	annotations only	annotations + (Yandex MT) output.
Avg. Understandability ratio (%)	19	26	28
Avg. Understandability decision time (sec)	20	80	150

From the evaluation results in Table 27, we see that, for the ‘jp-en’ language pair, the understandability ratio increased by 7% when the MT output was replaced by the AR annotations, and that it marginally increased by 2% when MT was added to AR.

The understandability decision time increased from 80 secs to 150 secs when moving from AR to AR+MT.

Synthesis .

The experimental results show that, for the “hi-en” and “jp-en” pairs, the ARO setting, the average understandability ratio increases by 5-7%, and further increases when evaluated jointly with MT output. With SUFT-1 as yet, this does not help us reach the understandability levels we desired.

Another interesting point is that the understandability decision time increases by about two times when the setting is changed from AR to AR+MT.

Conclusions and perspectives of the whole thesis

In this thesis, we proposed a tool-oriented active reading approach for the purpose of understanding tweets in foreign languages, especially Indian and Japanese tweets. We highlighted the needs for understanding foreign tweets in various contexts, and confirmed the inadequacies of two well-known freely available MT systems to tackle social media texts. In the particular case of tweets, we demonstrated the impossibility to improve MT systems to a good enough translation quality level. For that, we did a few preliminary experiments and observed that, at least for spontaneous tweets in Hindi and Japanese, the percentage of "understandable" tweets fell from 80% for native speakers of the source language (SL) to below 30% for English or French native speakers using MT and ignorant of the SL. That means that these large-scale MT systems make at least 62.5% (50/80) of understandable tweets non-understandable in the target language (TL). To get an *understandability ratio* of at least 50-60% on the whole set of tweets, we should lower the 62.5% ratio to 37.5% (30/80). That seems totally out of reach for any MT technique.

We then made some preliminary experiments to support our hypothesis that "multiple pidgin MT" presented in an Active Reading (AR) interface could immediately give a satisfactory understandability ratio, or at least lead towards a solution if sufficient lexical coverage could be put into play. To be able to prove that hypothesis and evaluate the various aspects of a SUFT (*System for Helping Understand Foreign Tweets*) based on it, we designed and implemented SUFT-1, a first prototype. We incorporated into it our subjective measure, the *understandability ratio* mentioned above (a tweet is judged as understandable or non-understandable), and added an objective measure, the *understandability decision time.*

Of course, we knew that such a system can be successful only if it has a large or very large lexical coverage. That is why we built or used various linguistic resources for Indian tweets: (1) monolingual dictionaries of forms precomputing the results of classical morphological parsers (as the DELAF for French)[83], that is, associating with them their possible lemmas and morphosyntactic features (POS, gender, number, case, person, tense…), and (2) bilingual dictionaries giving English equivalents and semantic descriptions[84].

An interesting aspect of our design is that it makes it possible, perhaps for the first time, to combine several morphological analyzers (in this case, for Hindi, Marathi and English) into a unique MA that can output all possible solutions into a compatibility graph, where, for example, a word form common to Hindi and Marathi can produce the union of the solutions for the two languages.

Another contribution is the design of a generic "SUFT" (System for helping Understand Foreign Tweets), as well as the specification and implementation of SUFT-1, an interactive multi-layout system based on AR, with a UI based on a browser to ensure portability, and easily configurable by adding dictionaries, morphological modules, and MT plugins. As for the layouts, we used M. Mangeot's CESSELIN "horizontal" interface, and, as the "tooltip" interface seemed to require too much knowledge of the SL for our evaluators, we designed and implemented our own "horizontal all-info" interface, that presents the possible solutions as a confusion graph obtained by calling MA and then a bilingual dictionary. The user can

[82] We used GOOGLE TRANSLATE and YANDEX, but our conclusions would be the same or worse for other systems.
[83] For this, we combined several open source lists compiled from word forms found in tweets: 163221 Hindi word forms corresponding to 68788 lemmas, and 72312 Marathi word forms corresponding to 6026 lemmas.
[84] For this, we used the HI-UNL CFILT dictionary containing about 136710 UWs, 65156 lemmas.

highlight a particular solution by selecting it, to guess a whole meaning by looking at the selected TL words. These solutions are presented in one "wrapped" line. The user can "push up" a solution (lemma + meaning + optionally SL grammatical features and pronunciation) in the horizontal layouts.

We performed experiments with SUFT-1 using a horizontal layout with 3 evaluation settings (1) (YANDEX MT) output only, (2) annotations only and (3) annotations + (YANDEX MT) output. For Japanese, the understandability ratio in SL was 90%, and for Indian tweets, it was 80%. The results can be summarized in the following table.

Table 28: Summary of evaluation results (il~Indian languages)

Setting / Language pair	(YANDEX MT) output only		annotations only		annotations + (YANDEX MT) output	
jp-en	19%	20s	26%	80s	28%	150s
jp-fr	12%	18s	6%	20s	8% (< 12%!)	25s
il-en	15%	14s	20%	70s	26%	140s

We had to discard the results for jp-fr as the evaluators did not "play the game": in particular, they declared less tweets to "make some sense" when they saw MT+AR than when they saw MT only.

On the positive side, considering experiments on jp-en and il-en, we see that the understandability ratio has gone up by about 5-6% when replacing the MT output by the AR annotations, and that showing MT and AR together still increase it. We thereby answer positively Question 1 and Question 2.

Did we really prove our point? No and yes! What comes out of these first experiments is that, although the understandability ratio obtained when using our AR interface only was higher than when showing MT results only, in the best setting, with 26%, we are far from our hopes (50-60%). Adding MT results gives a small improvement (2% for jp-en, 6% for il-en), still not enough by far to get the minimal 62.5% understandability ratio that we deemed necessary for a SUFT to be usable, meaning to be used (and not dropped after 5 minutes) and useful.

However, on the positive side, our experiments have shown that much progress can still be obtained, on several fronts.

1. Our lexical coverage is still limited, in particular for il-en, as the interplay between MA and bilingual dictionaries is not as good as it should be: word forms that should be replaced by lemmas are queried to the UNL dictionary as if they were lemmas.
2. The current layout of our AR presentation shows only 6—8 words with their annotations (pronunciation, lemma, POS, equivalents, features) in the main pane. We should modify it so that all information about a whole tweet appears together in it. We should also adapt to the user by letting him/her indicate some words as "well known", so that SUFT- would hide their annotations partially or totally.
3. We should also return to the idea of proposing a "vertical multiple layout", and find a way to modify the order of the TL elements of information contained in a word annotation. Indeed, in the current horizontal layout, they are contained in a vertical list where "pushing up" an element is easy.
   We abandoned our planned implementation of this type of interface because we did not find a type of interface element allowing to "move front" an element of a horizontal list, but we will investigate whether some widget of that kind has become available, and if not, we will consider to leave the elements in place, and to highlight the last "clicked" element of the list.

In the last month, we have implemented and experimented a "vertical multiple dynamic" interface along the lines above (previous point).

In parallel, we would like to redo our experiments on jp-fr with reliable evaluators of the desired profile, that is, interested in reading Japanese tweets and having an "advanced student" proficiency level in Japanese.

We also plan to improve the lexical coverage of our il-en data (for simple words), and then to use ATEF facilities for handling the portion of out of vocabulary (OOV) words that could be analyzed as compound words. In case a word is still not recognized (UNREC word), we plan to experiment a simple Levenshtein distance-based processing: generate all strings at distance one from the UNREC word and run the morphological analyzer and the bilingual lexicon on each of them[85]. That should handle some proportion of the many typing errors found in tweets.

Another interesting project is to use the TRADOH middleware of GETALP to call several MT servers in parallel, then to use recent quality estimation (QE) techniques to present the N best results in parallel with the AR annotations, and to evaluate improvements.

Last but not least, there has been a suggestion to try to use our AR multiple pidgin interface to help learners of foreign languages read e-books in those languages, when no good enough MT system is available, which is the case for the vast majority of under-resourced language pairs.

[85] Or, as has been proposed in the text above, precompute that by putting in some special dictionary (D6 in our case) all hypothetical word forms ϕ' that are at distance 1 from a word form ϕ appearing in our resource, and written in the same charset as ϕ.

Bibliography.

[1] Alansary, S., Nagi, M., & Adly, N. (2009). The universal networking language in action in English-Arabic machine translation. In *Proceedings of 9th Egyptian Society of Language Engineering Conference on Language Engineering,(ESOLEC 2009)* (pp. 23–24). Retrieved from http://www.alexandrina.org/isis/UploadedFiles/Publications/Cairo_2009.pdf

[2] André, P., Bernstein, M. S., & Luther, K. (2012). Who Gives A Tweet? Evaluating Microblog Content Value. In *Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work (CSCW 2012)* (pp. 471–474). Washington, USA. http://doi.org/10.1145/2145204.2145277. 9

[3] Bahramiana, Z., & Ali Abbaspoura, R. (2015). An ontology-based tourism recommender system based on Spreading Activation model. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives* (Vol. 40, pp. 83–90). http://doi.org/10.5194/isprsarchives-XL-1-W5-83-2015

[4] Bapat, M., Gune, H., & Bhattacharyya, P. (2010). A Paradigm-Based Finite State Morphological Analyzer for Marathi. In *Proceedings of the 1st Workshop on South and Southeast Asian Natural Language Processing (WSSANLP), colloc. with COLING 2010* (pp. 26–34). Beijing. Retrieved from https://www.cse.iitb.ac.in/~pb/papers/coling10-wssanlp-marathi-ma.pdf

[5] Bergsma, S., McNamee, P., Bagdouri, M., Fink, C., & Wilson, T. (2012). Language Identification for Creating Language-Specific Twitter Collections. In *Proceedings of the 2012 Workshop on Language in Social Media (LSM 2012)* (pp. 65–74). Montreal, Canada: Association for Computational Linguistics. Retrieved from http://www.aclweb.org/anthology-new/W/W12/W12-2108.pdf

[6] Bhattacharyya, P. (2014). Facilitating Multi-Lingual Sense Annotation : Human Mediated Lemmatizer. In *Global Wordnet Conference*. Tartu, Estonia.

[7] Bögel, T., Butt, M., Hautli, A., & Sulger, S. (2007). Developing a finite-state morphological analyzer for Urdu and Hindi. *Finite State Methods and Natural Language Processing*, 10. Retrieved from http://ling.uni-konstanz.de/pages/home/butt/main/papers/boegeletal.pdf

[8] Boitet, C. (1997). GETA's MT methodology and its current development towards personal networking communication and speech translation in the context of the UNL and C-STAR projects. In *Proceedings of the Conference of the Pacific Association for Computational Linguistics (PACLING)* (p. 35). Retrieved from http://www-clips.imag.fr/geta/christian.boitet/pages_personnelles/zArticles_sur_la_TAO_pdf/Pacling.CBinv.v4.pdf

[9] Boitet, C., Bellynck, V., Mangeot, M., & Ramisch, C. (2008). Towards Higher Quality Internal and Outside Multilingualization of Web Sites. In *Proceedings of the Summer Workshop on Ontology, NLP, Personalization and IE/IR (ONII-08)* (p. 8). Bangalore, India.

[10] Boitet, C., Blanchon, H., Seligman, M., & Bellynck, V. (2009). Evolution of MT with the Web. In *Proceedings of the International Conference "Machine Translation 25 Years On"* (pp. 1–13). Cranfield, England.

[11] Boitet, C., Mangeot, M., & Sérasset, G. (2002). The PAPILLON project: Cooperatively building a multilingual lexical data-base to derive open source dictionaries and lexicons. In *Proceedings of the 2nd Workshop on NLP and XML at COLING'02* (pp. 93–96). Taipei, Taiwan. http://doi.org/hal-00965824

[12] Bostandjiev, S., Donovan, J. O., & Höllerer, T. (2012). TasteWeights: A visual interactive hybrid recommender system. In *Proceedings of the 6th ACM conference on Recommender systems* (pp. 35–42). http://doi.org/10.1145/2365952.2365964

[13] Burke, R. (2007). Hybrid web recommender systems. *The Adaptive Web*, 377–408. http://doi.org/10.1007/978-3-540-72079-9_12

[14] Chandioux, J. (1989). 10 ans de METEO (MD). In A. Abbou (Ed.), *Proceedings of Traduction Assistée par Ordinateur: Perspectives technologiques, industrielles et économiques envisageables à l'Horizon 1990: l'offre, la demande, les marchés et les évolutions en cours* (pp. 169–172). Paris, France: Daicadif.

[15] Chatterjee, A., Joshi, S. R., Khapra, M. M., & Bhattacharyya, P. (2010). Introduction to Tools for IndoWordNet and Word Sense Disambiguation. In *In Proceedings of 3rd IndoWordNet workshop, ICON.*

[16] Chauché, J. (1975). The ATEF and CETA systems. In D. Hays (Ed.), *American Journal of Computational Linguistics (2, microfiche 17)* (pp. 21–40). Association for Computational Linguistics.

[17] Chen, J., Nairn, R., Nelson, L., Bernstein, M., & Chi, E. (2010). Short and tweet: experiments on recommending content from information streams. In *SIGCHI Conference on Human Factors in Computing Systems* (pp. 1185–1194). http://doi.org/10.1145/1753326.1753503

[18] Chen, Y., Wang, L., Boitet, C., & Shi, X. (2014). On-going Cooperative Research towards Developing Economy-Oriented Chinese-French SMT Systems with a New SMT Framework. In *Proceedings of 21st Traitement Automatique des Langues Naturelles, TALN* (pp. 401–406). Marseille, France.

[19] Chittaranjan, G., Vyas, Y., Bali, K., & Choudhury, M. (2014). Word-level Language Identification using CRF : Code-switching Shared Task Report of MSR India System. In *Proceedings of The First Workshop on Computational Approaches to Code Switching at EMNLP'14* (pp. 73–79). Doha, Qatar.

[20] Dabre, R., Amberkar, A., & Bhattacharyya, P. (2012). Morphological Analyzer for Affix Stacking Languages: A Case Study of Marathi. In *Proceedings of the 24th International Conference on Computational Linguistics* (p. 9). Retrieved from https://www.cse.iitb.ac.in/~pb/papers/coling12-marathi-morph-analyser.pdf

[21] Daoud, M. (2010). *Usage of non-conventional resources and contributive methods to bridge the terminological gap between languages by developing multilingual "preterminologies."* (Doctoral dissertation, University of Grenoble), Grenoble, France.

[22] Depraetere, H., & Van de Walle, J. (2012). Bologna Translation Service: An enabler for international profiling and student mobility. In *Proceedings of the 6th International Technology, Education and Development Conference* (pp. 5907–5912). Valencia, Spain: IATED.

[23] Depraetere, H., Van den Bogaert, J., & Van de Walle, J. (2011). Bologna Translation Service: Online translation of course syllabi and study programmes in English. In *Proceedings of the 15th Conference of the European Association for Machine Translation* (pp. 29–34). Leuven, Belgium.

[24] Derczynski, L., Ritter, A., Clark, S., & Bontcheva, K. (2013). Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *Proceedings of the Recent Advances in Natural Language Processing* (pp. 198–206). Hissar, Bulgaria. Retrieved from http://www.aclweb.org/website/old_anthology/R/R13/R13-1026.pdf

[25] Desai, U., & Ramsay-Brijball, M. (2004). Tracing Gujarati Language Development Philologically and Sociolinguistically. *Alternation: International Journal for the Study of Southern African Literature and Languages*, *11*(2), 308–324.

[26] Devi, R. G., Veena, P. V, Kumar, A. M., & Soman, K. P. (2016). AMRITA_CEN@FIRE 2016: Code-Mix Entity Extraction for Hindi-English and Tamil-English Tweets. In *Forum for Information Retrieval and Evaluation* (p. 5).

[27] Dey, A., & Fung, P. (2014). A Hindi-English Code-Switching Corpus Code-Switching in Indian Culture. In *Proceedings of the 9th Language Resources and Evaluation Conference* (pp. 2410–2413).

[28] Ekstrand, M. D., Riedl, J. T., & Konstan, J. A. (2010). Collaborative Filtering Recommender Systems. *Foundations and Trends® in Human–Computer Interaction*, *4*(2), 81–173. http://doi.org/10.1561/1100000009

[29] Falaise, A., Rouquet, D., Schwab, D., Blanchon, H., & Boitet, C. (2010). Ontology driven content extraction using interlingual annotation of texts in the OMNIA project. In *Proceedings of the 4th International Workshop on Cross-Lingual Information Access at COLING'10* (pp. 52–60). Beijing, China.

[30] Farzindar, A., & Inkpen, D. (2015). *Natural Language Processing for Social Media*. (Graeme Hirst, Ed.)*Natural Language Processing for Social Media* (8(2), Vol. 8). Morgan and Claypool. http://doi.org/10.2200/S00659ED1V01Y201508HLT030

[31] Finin, T., Murnane, W., Karandikar, A., Keller, N., Martineau, J., & Dredze, M. (2010). Annotating Named Entities in Twitter Data with Crowdsourcing. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk* (Vol. 2010, pp. 80–88). Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.169.4068&rep=rep1&type=pdf%5Cn http://www.aclweb.org/anthology/W10-0713

[32] Fleischman, M., & Hovy, E. (2003). Recommendations without user preferences: a natural language processing approach. In *Proceedings of the 8th International Conference on Intelligent User Interfaces* (pp. 242–244). http://doi.org/10.1145/604045.604087

[33] Francopoulo, G., Bel, N., George, M., Calzolari, N., Monachini, M., Pet, M., & Soria, C. (2009). Multilingual resources for NLP in the lexical markup framework (LMF). In *Language Resources and Evaluation* (Vol. 43, pp. 57–70). http://doi.org/10.1007/s10579-008-9077-5

[34] Gambäck, B., Eriksson, G., & Fourla, A. (2005). Natural Language Processing at the School of Information Studies for Africa. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*. Retrieved from www.sics.se/$~$gamback/publications/acl05-teachnlp.pdf

[35] Gauthier, M., Guille, A., Rico, F., & Deseille, A. (2015). Text mining and Twitter to analyze British swearing habits. In C. Levallois, M. Marchand, T. Mata, & A. Panisson (Eds.), *Proceedings of International Conference on Twitter for Research* (pp. 28–42). Lyon, France: EMLYON Press. Retrieved from https://www.researchgate.net/publication/291321112_Handbook_Twitter_for_Research_2015_2016

[36] Gimpel, K., Schneider, N., O'Connor, B., Das, D., Mills, D., Eisenstein, J., … Smith, N. A. (2011). Part-of-Speech Tagging for Twitter: Annotation, Features, and Experiments. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Shortpapers*, (2), 42–47. http://doi.org/10.1.1.206.3224

[37] Goldsmith, J. (2001). Unsupervised Learning of the Morphology of a Natural Language. *Computational Linguistics*, *27*(2), 153–198. http://doi.org/10.1162/089120101750300490

[38] Goldsmith, J. (2006). An algorithm for the unsupervised learning of morphology. *Natural Language Engineering*, *12*(4), 353. http://doi.org/10.1017/S1351324905004055

[39] Gotti, F., Langlais, P., & Farzindar, A. (2013). Translating Government Agencies' Tweet Feeds: Specificities, Problems and (a few) Solutions, 80–89.

[40] Gotti, F., Langlais, P., & Farzindar, A. (2014). Hashtag Occurrences, Layout and Translation: A Corpus-driven Analysis of Tweets Published by the Canadian Government. In *Proceedings of the 9th Language Resources and Evaluation Conference* (pp. 2254–2261). Retrieved from http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:No+Title#0

[41] Goyal, P., Mital, M. R., Mukerjee, A., Raina, A. M., Sharma, D., Shukla, P., & Vikram, K. (2003). A bilingual parser for Hindi, English and code-switching structures. In *Proceedings of the European Association for Computational LInguistics* (p. 15). Retrieved from https://www.cs.cornell.edu/~kvikram/papers/eacl2003.pdf

[42] Green, S., Heer, J., & Manning, C. D. (2013). The efficacy of human post-editing for language translation. In *Proceedings of the SIGCHI conference on Human factors in Computing systems* (pp. 439–448).

[43] Guilbaud, J.-P., Boitet, C., & Berment, V. (2013). Un analyseur morphologique étendu de l'allemand traitant les formes verbales à particule séparée. In *Proceedings of TALN-RÉCITAL* (Vol. 9, pp. 755–763). Les Sables d'Olonne, France.

[44] Habash, N. Y. (2010). Introduction to Arabic Natural Language Processing. In G. Hirst (Ed.), *Synthesis Lectures on Human Language Technologies* (Vol. 3, pp. 1–187). Toronto: Morgan and Claypool. http://doi.org/10.2200/S00277ED1V01Y201008HLT010

[45] Han, B., Cook, P., Au, P. E., & Baldwin, T. (2014). Text-Based Twitter User Geolocation Prediction. *Journal of Artificial Intelligence Research*, *49*, 451–500. Retrieved from http://jair.org/media/4200/live-4200-7781-jair.pdf

[46] Hannon, J., Bennett, M., & Smyth, B. (2010). Recommending Twitter Users to Follow Using Content and Collaborative Filtering Approaches. In *Proceedings of the 4th Conference on Recommender Systems* (pp. 199–206). Barcelona, Spain. http://doi.org/10.1145/1864708.1864746

[47] Harris, B. (1976). The Importance of Natural Translation. In *Working Papers in Bilingualism* (pp. 96–114). Toronto. Retrieved from https://www.academia.edu/Documents/in/Brian_harris_the_importance_of_natural_translation_Published_in_Working_Papers_in_Bilingualism

[48] Harris, B., & Hofmann, T. (1970). Pidgin Translation. *Meta*, *15*(2), 71–87.

[49] Huynh, C.-P. (2010). *Des suites de test pour la TA à un système d'exploitation de corpus alignés de documents et métadocuments multilingues, multiannotés et multimédia*. (Doctoral dissertation, Université Grenoble-Alpes). Retrieved from https://tel.archives-ouvertes.fr/tel-00548196/document

[50] Huynh, C.-P., Boitet, C., & Blanchon, H. (2008). SECTra_w.1: An online collaborative system for evaluating, post-editing and presenting MT translation corpora. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation* (pp. 2571–2576). Retrieved from http://mt-archive.info/LREC-2008-Huynh.pdf%5Cnpapers2://publication/uuid/F74BC6E9-3328-4450-83A1-43DC4F4B7FC7

[51] Isabelle, P. (1987). Machine Translation at the TAUM group. In *Proceedings of Machine Translation Today: The State of the Art* (Margaret K, pp. 247–277). Edinburgh: Edinburgh University Press.

[52] Isahara, H. (1995). JEIDA's test-sets for quality evaluation of MT systems–technical evaluation from the developer's point of view. In *Proceedings of MT Summit V* (p. 10). Retrieved from http://mt-archive.info/MTS-1995-Isahara.pdf

[53] Jehl, L. E. (2010). Machine Translation for Twitter. Retrieved from http://hdl.handle.net/1842/5317

[54] Jehl, L., Hieber, F., & Riezler, S. (2012). Twitter Translation using Translation-Based Cross-Lingual Retrieval, 410–421.

[55] Kalitvianski, R., Boitet, C., & Bellynck, V. (2012). Collaborative computer-assisted translation applied to pedagogical documents and literary works. In *Proceedings of the 24th International Conference on Computational Linguistics* (pp. 255–260). Mumbai, India.

[56] Kim, S., Weber, I., Wei, L., & Oh, A. (2014). Sociolinguistic Analysis of Twitter in Multilingual Societies. In *Proceedings of the 25th ACM conference on Hypertext and Social Media* (pp. 243–248). New York, US: ACM Press. http://doi.org/10.1145/2631775.2631824

[57] Krishn, A., Guha, R. S., & Mukherjee, A. (2012). *Unsupervised Morphological Analysis of Hindi*. (Project report, Indian Institute of Kanpur), Kanpur, India. Retrieved from http://www.cse.iitk.ac.in/users/cs365/2012/submissions/rabisg/cs365/projects/report.pdf

[58] Kumar, G. H., & Seyedmahmoud, T. (2016). Movie Recommendation based on Users ' Tweets. In *International Journal of Computer Applications* (Vol. 141, pp. 34–36).

[59] Kywe, S. M., Lim, E.-P., & Zhu, F. (2012). A Survey of Recommender Systems in Twitter. In *Proceedings of the Research Collection School Of Information Systems* (Vol. 7710, pp. 420–433). http://doi.org/10.1007/978-3-642-35386-4_31

[60] L. Gavrilova. (2006). Analysis of Users' Interest Based on Tweets. *Computational Science and Its Applications*, *1*, 354. http://doi.org/10.1007/978-3-642-12179-1

[61] Lafourcade, M., & Chauché, J. (1998). Ficus - un agent dictionnaire coopératif et extensible. In *NLP+IA'98* (p. 8). Moncton, New Brunswick, Canada.

[62] Larsen, J. B. (2013). *Content-based Recommender Systems*. (Doctoral dissertation, Technical University of Denmark), Copenhagen, Denmark.

[63] Ling, W., Xiang, G., Dyer, C., Black, A., & Trancoso, I. (2013). Microblogs as Parallel Corpora. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics* (pp. 176–186). Sofia, Bulgaria. Retrieved from http://aclweb.org/anthology/P13-1018

[64] Lops, P., Gemmis, M. De, & Semeraro, G. (2011). Content-based Recommender Systems: State of the Art and Trends. *Recommender Systems Handbook*, 73–105. Retrieved from http://link.springer.com/chapter/10.1007/978-0-387-85820-3_3

[65] Lui, M., & Baldwin, T. (2014). Accurate Language Identification of Twitter Messages. *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)@EACL 2014*, 17–25.

[66] Mahmud, J., Nichols, J., & Drews, C. (2012). Where Is This Tweet From? Inferring Home Locations of Twitter Users. In *Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media Where* (pp. 511–514). http://doi.org/papers3://publication/uuid/8AAE166A-DE81-42AB-83BC-5E014B7B0039

[67] Mahmud, J., Nichols, J., & Drews, C. (2014). Home Location Identification of Twitter Users. *ACM Transactions on Intelligent Systems and Technology*, *5*(3), 47–69. http://doi.org/10.1145/2528548

[68] Malik, A., Boitet, C., & Bhattacharyya, P. (2008). Hindi Urdu Machine Transliteration using Finite-state Transducers. In *Proceedings of the 22nd International Conference on Computational Linguistics* (pp. 537–544). Manchester, US. http://doi.org/10.3115/1599081.1599149

[69] Malmivuo, J., & Plonsey, R. (1995). *Bioelectromagnetism: principles and applications of bioelectric and biomagnetic fields*. Oxford University Press. Retrieved from http://www.bem.fi/book/

[70] Mangeot-Nagata, M. (2016). Collaborative Construction of a Good Quality, Broad Coverage and Copyright Free Japanese- French Dictionary. *Internation Journal of Lexicography*, 30. Retrieved from http://doi.org/10.1093/ijl/ecw035

[71] Mehrotra, R., Sanner, S., Buntine, W., & Xie, L. (2013). Improving LDA Topic Models for Microblogs via Tweet Pooling and Automatic Labeling. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 889–892). http://doi.org/10.1145/2484028.2484166

[72] Melville, P., Mooney, R. J., & Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations. In *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI)* (pp. 187–192). Edmonton, Canada. http://doi.org/10.1.1.16.4936

[73] Mesthrie, R. (1997). Indian Languages in Africa: a field report, 1900-1995. *Yearbook of South Asian Languages*. Retrieved from http://astiane.com/view/language-in-south-africa-rajend-mesthrie-pdf

[74] Mukherjee, S., Malu, A., Balamurali, A. R., & Bhattacharyya, P. (2012). TwiSent: A Multistage System for Analyzing Sentiment. In *Proceedings of the Conference on Information and Knowledge Management (CIKM)* (pp. 2531–2534). http://doi.org/10.1145/2396761.2398684

[75] Nagao, M., Tsujii, J., & Nakamura, J. (1985). The Japanese Government Project for Machine Translation. *Computational Linguistics*, *11*(2–3), 91–110.

[76] Nastase, V., & Strube, M. (2008). Decoding Wikipedia Categories for Knowledge Acquisition. In *Proceedings of the 23rd national conference on Artificial intelligence* (pp. 1219–1224). Retrieved from http://dl.acm.org/citation.cfm?id=1620163.1620262

[77] Nayan, A., Rao, B. R. K., Singh, P., Sanyal, S., & Sanyal, R. (2008). "Named Entity Recognition for Indian Languages." In *Proceedings of the IJCNLP-08 Workshop on NER for South and South East Asian Languages* (Vol. 3, pp. 97–104).

[78] Neubig, G., & Duh, K. (2013). How Much Is Said in a Tweet? A Multilingual, Information-theoretic Perspective. In *AAAI Spring Symposium: Analyzing Microtext* (pp. 32–39). Retrieved from http://www.aaai.org/ocs/index.php/SSS/SSS13/paper/viewFile/5698/5906

[79] Nguyen, D. P. T., & Ishizuka, M. (2006). A statistical approach for universal networking language-based relation extraction. In *Proceedings of the 4th IEEE International Conference on Research, Innovation and Vision for the Future, RIVF'06* (pp. 153–160). IEEE. http://doi.org/10.1109/RIVF.2006.1696432

[80] Nguyen, H.-T., Boitet, C., & Sérasset, G. (2007). PIVAX, an online contributive lexical database for heterogeneous MT systems using a lexical pivot. In *Proceedings of the International Symposium on Natural Language Processing*. Bangkok, Thailand.

[81] Nguyen, M.-T., Kitamoto, A., & Nguyen, T.-T. (2015). TSum4act: A Framework for Retrieving and Summarizing Actionable Tweets during a Disaster for Reaction. In *The 19th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2015)* (Vol. 9078, p. 12). Ho Chi Minh City, Vietnam. http://doi.org/10.1007/978-3-319-18032-8

[82] Nikolova, B., & Nenova, I. (1982). An Automated System for Term Services. In J. Horeckp (Ed.), *Proceedings of the 9th Conference on Computational Linguistics* (pp. 265–270). Prague, Czechoslovakia: North-Holland.

[83] Nomura, H., & Isahara, H. (1992). Evaluation Surveys: The JEIDA Methodology and Survey. In *MT Evaluation: Basis for Future Directions, Proceedings of a workshop sponsored by the National Science Foundation* (pp. 11–12). Retrieved from http://www.mt-archive.info/AMTA-1992-Nomura.pdf

[84] Oostdijk, N. (2015). MapTwitter tribal language(s). In C. Levallois, M. Marchand, T. Mata, & A. Panisson (Eds.), *Proceedings of International Conference on Twitter for Research* (pp. 65–87). Lyon, France: EMLYON Press. Retrieved from https://www.researchgate.net/publication/291321112_Handbook_Twitter_for_Research_2015_2016

[85] Pak, A., & Paroubek, P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In *Proceedings of the Seventh Conference on International Language Resources and Evaluation* (pp. 1320–1326). http://doi.org/10.1371/journal.pone.0026624

[86] Park, D., Kim, H., Choi, I., & Kim, J. (2012). A Literature Review and Classification of Recommender Systems on Academic Journals. *Expert Systems with Applications*, *39*(11), 139–152. Retrieved from http://jiisonline.evehost.co.kr/files/DLA/9-17-1.pdf

[87] Patawar, M., & Potey, M. (2016). Named Entity Recognition from Indian tweets using Conditional Random Fields based Approach. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, *5*(5), 5p.

[88] Patel, R. B. (1967). Etymological and Phonetic Changes among Foreign words in Kiswahili. *Journal of the Institute of Swahili Research*, *37*(1), 6p. Retrieved from http://files.eric.ed.gov/fulltext/ED023068.pdf

[89] Pennacchiotti, M., & Popescu, A.-M. (2011). A Machine Learning Approach to Twitter User Classification. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media* (pp. 281–288). Retrieved from http://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/viewFile/2886/3262

[90] Pietrzak, J., Jauregi, A., Van de Walle, J., & Eriksson, A. (2013). Improving access to educational courses via automatic machine translation - New developments in post-editing. In *Proceedings of the INTED 2013 Conference* (pp. 5521–5529). IATED.

[91] Pouliquen, B., Elizalde, C., Junczys-Dowmunt, M., Mazenc, C., & García-Verdugo, J. (2013). Large-scale multiple language translation accelerator at the United Nations. In *Proceedings of the 14th Machine Translation Summit* (pp. 345–352).

[92] Pouliquen, B., & Mazenc, C. (2011a). Automatic translation tools at WIPO. *ASLIB, Translating and the Computer*, *33*, 17–18.

[93] Pouliquen, B., & Mazenc, C. (2011b). COPPA, CLIR and TAPTA: Three tools to assist in overcoming the patent language barrier at WIPO. In *Proceedings of the 13th Machine Translation Summit* (pp. 24–30).

[94] Probyn, J. (2016). Study sheds light on how Africans use Twitter. Retrieved from http://www.howwemadeitinafrica.com/study-sheds-light-africans-use-twitter/54249/

[95] Ramage, D., Dumais, S., & Liebling, D. (2010). Characterizing Microblogs with Topic Models. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media Characterizing* (pp. 1–8).

[96] Ramisch, C., Villavicencio, A., & Boitet, C. (2010). Web-based and combined language models: a case study on noun compound identification. In *Proceedings of the 23rd International Conference on Computational Linguistics* (pp. 1041–1049). Beijing, China. Retrieved from http://aclweb.org/anthology-new/C/C10/C10-2120.pdf

[97] Ritter, A., Clark, S., & Etzioni, O. (2011). Named entity recognition in tweets: an experimental study. *... of the Conference on Empirical Methods ...*, 1524–1534. http://doi.org/10.1075/li.30.1.03nad

[98] Rolinger, L. (2009). *Edible Identities: Food, Cultural Mixing and the Making of Identities on the Swahili coast*. University of Alberta.

[99] Rouquet, D., & Nguyen, H.-T. (2009). Interlingual annotation of texts in the OMNIA project. In *Proceedings of the 4th Language and Technology Conference* (pp. 290–294). Poznan, Poland.

[100] Saha, S. K., Sarkar, S., & Mitra, P. (2008). "Gazetteer preparation for named entity recognition in Indian languages." In *6th Workshop on Asian Language Resources collocated with International Joint Conference on Natural Language Processing* (pp. 9–16). Retrieved from http://pascasarjana.mercubuana.ac.id/28/I08-7.pdf#page=17

[101] Saito, I., Sadamitsu, K., Asano, H., & Matsuo, Y. (2014). Morphological Analysis for Japanese Noisy Text based on Character-level and Word-level Normalization. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING): Technical Papers* (pp. 1773–1782). Retrieved from http://www.aclweb.org/anthology/C14-1167

[102] Sankaran, B., Sankaran, B., Bali, K., Bali, K., Choudhury, M., Choudhury, M., … Subbarao, K. V. (2008). A Common Parts-of-Speech Tagset Framework for Indian Languages. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, (January), 1331–1337. Retrieved from http://www.lrec-conf.org/proceedings/lrec2008/

[103] Sasano, R., Kurohashi, S., & Okumura, M. (2014). A Simple Approach to Unknown Word Processing in Japanese Morphological Analysis (in {Japanese}). *Journal of Natural Language Processing*, *21*(6), 1183–1205. http://doi.org/10.5715/jnlp.21.1183

[104] Scannell, K. P. (2007). The Crúbadán Project: Corpus building for under-resourced languages. In *Building and Exploring Web Corpora: Proceedings of the 3rd Web as Corpus Workshop* (Vol. 4, pp. 5–15). Retrieved from http://borel.slu.edu/pub/wac3.pdf

[105] Seligman, M., Boitet, C., & Meddeb-Hamrouni, B. (1998). Transforming lattices into non-deterministic automata with optional null arcs. In *Proceedings of the 36th annual meeting on Association for Computational Linguistics* - (Vol. 2, pp. 1205–1211). http://doi.org/10.3115/980691.980766

[106] Shah, R. (2016). *Resource collection in the form of dictionaries, named-entities and tweets in Hindi and Marathi: Experiments for building a recommender using tweets*. (Internship report, National Institute of Informatics), Tokyo, Japan.

[107] Shah, R., & Boitet, C. (2015). Understandability of machine-translated Hindi tweets before and after post-editing: Perspectives for a recommender system. *CEUR Workshop Proceedings*, *1445*, 44–50.

[108] Shah, R., Boitet, C., & Bhattacharyya, P. (2015). Building a recommender system using multilingual multiscript tweets. In *Proceedings of the International Conference on Twitter Research (CTR)* (pp. 160–170). Lyon, France.

[109] Sharma, R., Gupta, M., Agarwal, A., & Bhattacharyya, P. (2015). Adjective Intensity and Sentiment Analysis. In *Proceedings of Conference on Empirical Methods in Natural Language Processing*. Lisboa, Portugal.

[110] Si, A. (2011). A diachronic investigation of Hindi-English code-switching, using Bollywood film scripts. *International Journal of Bilingualism*, *15*(4), 388–407. http://doi.org/10.1177/1367006910379300

[111] Singh, S., & Sarma, V. (2011). Verbal Inflection in Hindi: A Distributed Morphology Approach. In *Proceedings of 25th Pacific Asia Conference on Language, Information and Computation* (pp. 283–292).

[112] Tchechmedjiev, A., Goulian, J., Schwab, D., & Sérasset, G. (2012). Parameter estimation under uncertainty with Simulated Annealing applied to an ant colony based probabilistic WSD algorithm. In *"Proceedings of the First International Workshop on Optimization Techniques for Human Language Technology"* (pp. 109–124). Retrieved from http://www.aclweb.org/anthology/W12-6108

[113] Terdalkar, H., & Agarwal, S. (2015). Romanagari Detection in Twitter (pp. 1–13). Kanpur, India.

[114] Tiedemann, J. (2012). Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC)* (pp. 2214–2218). Retrieved from http://lrec.elra.info/proceedings/lrec2012/pdf/463_Paper.pdf

[115] Tiedemann, J., & Nygaard, L. (2003). OPUS - an open source parallel corpus. In *"Proceedings of 14th Nordic Conference on Computational Linguistics (NoDaLiDa)"* (pp. 1–8). Reykjavik,

Iceland. Retrieved from http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:OPUS+-+an+open+source+parallel+corpus#2

[116] Tiedemann, J., & Nygaard, L. (2004). The OPUS corpus - parallel & free. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC)* (Vol. 22, p. 4). Lisbon, Portugal. http://doi.org/10.2307/1483710

[117] Tromp, E., & Pechenizkiy, M. (2011). Graph-based N-gram language identification on short texts. In *Proceedings of the 20th annual Belgian-Dutch Conference on Machine Learning* (pp. 27–34).

[118] Van de Walle, J., Depraetere, H., & Pietrzak, J. (2012a). Bologna Translation Service: High-quality automated translation of study programmes into English. In *"Proceedings of the EDULEARN 2012 Conference"* (pp. 5831–5835). IATED.

[119] Van de Walle, J., Depraetere, H., & Pietrzak, J. (2012b). Bologna Translation Service: Making study programmes accessible throughout Europe by means of high-quality automated translation. In *"Proceedings of ICERI 2012 Conference"* (pp. 3910–3918). IATED.

[120] van Halteren, H. (2008). Source Language Markers in EUROPARL Translations. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING)* (pp. 937–944). Manchester, UK. http://doi.org/10.3115/1599081.1599199

[121] Vauquois, B., & Boitet, C. (1985). Automated translation at Grenoble University. *Computational Linguistics*, *11*(1), 28–36.

[122] Virpioja, S., Smit, P., Grönroos, S.-A., & Kurimo, M. (2013). *Morfessor 2.0: Python Implementation and Extensions for Morfessor Baseline* (Aalto Univ). Department of Signal Processing and Acoustics, Aalto University. Retrieved from http://urn.fi/URN:ISBN:978-952-60-5501-5

[123] Vogel, J., & Tresner-kirsch, D. (2012). Robust Language Identification in Short, Noisy Texts : Improvements to LIGA. In *Proceedings of the 3rd International Workshop on Mining Ubiquitous and Social Environments* (pp. 1–9).

[124] Vyas, Y., Gella, S., Sharma, J., Bali, K., & Choudhury, M. (2014). POS Tagging of English-Hindi Code-Mixed Social Media Content. In A. for C. Linguistics (Ed.), *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 974–979). Doha, Qatar.

[125] Wang, L., & Boitet, C. (2013). Online production of HQ parallel corpora and permanent task-based evaluation of multiple MT systems: both can be obtained through iMAGs with no added cost. In *Proceedings of the 2nd Workshop on Post-Editing Technologies and Practice at MT Summit 2013* (pp. 103–110). Nice, France.

[126] Wang, Z., & Iwaihara, M. (2015). Cross-lingual Tweet Recommendation Based on User Interest Using Bilingual LDA Related work. In *Proceedings of 7th Forum on Data Engineering and Information Management* (p. 8). Retrieved from http://db-event.jpn.org/deim2015/paper/21.pdf

[127] Yan, R., & Li, X. (2012). Tweet Recommendation with Graph Co-Ranking. *Jeju, Republic of Korea*, 516–525. Retrieved from http://www.aclweb.org/anthology/P12-1054

[128] Yang, S., & Kavanaugh, A. L. (2011). Half-Day Tutorial : Collecting, Analyzing and Visualizing Tweets using Open Source Tools. In *Proceedings of the 12th Annual International Conference on Digital Government Research* (pp. 374–375). http://doi.org/10.1145/2037556.2037633

[129] Yang, X., Guo, Y., Liu, Y., & Steck, H. (2014). A survey of collaborative filtering based social recommender systems. *Computer Communications*, *41*, 1–10. http://doi.org/10.1016/j.comcom.2013.06.009

[130] Ying, Z. (2016). *Modèles et outils pour des bases lexicales "métier" multilingues et contributives de grande taille, utilisables tant en traduction automatique et automatisée que pour des services dictionnairiques variés*. (Doctoral dissertation, Université Grenoble-Alpes), Grenoble, France.

[131] Zielinski, A., & Bügel, U. (2012). Multilingual Analysis of Twitter News in Support of Mass Emergency Events. In *Proceedings of International Conference on Information Systems for Crisis Response and Management (ISCRAM)* (Vol. 5, pp. 5–10). http://doi.org/10.3233/SW-130110

Recapitulation of definitions

Definition 1: Understandability ratio.
The understandability ratio is the percentage of tweets that are understandable by a user of a certain profile, in a certain context, e.g. using the source tweets only, MT results, or Active Reading aid.

Definition 2: Understandability decision time.
The Understandability decision time is the average time it takes for a user of a certain profile, in a certain context (e.g. using the source tweets only, MT results, or Active Reading aid) to decide that s/he can "make sense of it" or not.

Definition 3: Code-mixing.
This qualifies spoken or written utterances containing words of more than one language or dialect.

Recapitulation of questions on factors influencing UFTweets

Question 1: Does Active Reading really improve understandability of foreign tweets, and if so by how much?

Question 2: Is it useful to show an MT proposal alongside an Active Reading presentation?

Question 3: What can be done in a SUFT in the case of OOV words?

Question 4: If we incorporate NEs in the AR module, will it help better elicit the context of the tweet or the tweet translation?

Question 5: Will the incorporation of NEs in the AR module help get around the problem of the large vocabulary coverage inherent in the tweets?

Question 6: How to measure whether SUFT would be useful for also helping people who want to progress in their knowledge of the SL?

We answered them as follows.

Question 1: Yes.

Question 2: Yes, but marginally, and then the understandability decision time increases by about 100% (it doubled in our experiments).

Question 3: We described 2 complementary approaches, but did not implement them yet.

Question 4: We extracted NEs and made some introspective evaluations. We are sure the answer is yes, but cannot quantify this result yet.

Question 5: Yes, but it will only be part of the solution.

Question 6: We don't know yet. Perhaps one could begin by testing these learners on tweets. But then, there should be a question on how well they think they understood.

Appendices

Appendix 1 Semantically related query terms collected using a Hindi synset

भोजनालय	मंगल उत्सव	उद्घाटन समारोह
रेस्तराँ	झूलन	उद्घाटन
रेस्त्रां	हिंडोला	नृत्य समारोह
रेस्तरा	हिन्डोला	डांस पार्टी
होटल	जयंती	ओलंपिक खेल
रेस्तरां	जयन्ती	ओलम्पिक खेल
रेस्टोरेंट	महोत्सव	भव्य समारोह
रेस्टोरेन्ट	वसंतोत्सव	विवाह समारोह
रेस्टरांट	वसंत उत्सव	विवाह
रेस्टरान्ट	मदनोत्सव	शादी
ढाबा	मदन-महोत्सव	शादी-ब्याह
धाबा	वसंत-महोत्सव	वैवाहिक
बासा	वार्षिकोत्सव	ग्राम्म
सराय	वार्षिक समारोह	इंद्रध्वज
मुसाफिरखाना	वार्षिक उत्सव	इन्द्रध्वज
मुसाफ़िर	जलसा	इंद्र-ध्वज
ख़ाना	जल्सा	इन्द्र-ध्वज
पांथशाला,	महफ़िल	इंद्रध्वज
पथिकालय	महफिल	इन्द्रध्वज
पथिकाश्रय	मजलिस	इंद्र-ध्वज
जनाश्रय	नगर कीर्तन	इन्द्र-ध्वज
लॉज	नगर कीरतन	रतजगा
लाज	रास	सम्मान समारोह
धर्मशाला	गणेशोत्सव	सत्कार समारोह
धर्म-शाला	गणेश	सम्मानोत्सव
जनाश्रय	उत्सव	फिल्मोत्सव
उत्सव	दुर्गोत्सव	फ़िल्मोत्सव
समारोह,	अन्नकूट	फिल्म उत्सव
मंगलोत्सव	अन्न-कूट	फिल्म महोत्सव
शुभोत्सव	मुहूर्त	एकसठी
शुभ उत्सव	उद्घाटन-समारोह	

Appendix 2 Experiments concerning Gujarati tweets from Africa

1. Details on queries submitted and results obtained

a. 'lang :gu' as a query

f1 {lang:gu},10000,15-06-2015-1253

f3 {lang:gu},265,06-05-2016-1449

f4 {lang:gu},2895,02-11-2016-1801

f6 {lang:gu},392,02-05-2016-1807

f7 {lang:gu},396,07-05-2016-1617

b. 5 Gujarati unigrams {ane (and), HatA (were), cho (are), chuM (am), che (is)} as queries

f197 {અને},780,08-05-2016-1709

f153 {હતા},484,06-05-2016-1601

f58 {છો},14,06-05-2016-2123

f123 {છું},343,06-05-2016-2124

f71 {છે},1778,06-05-2016-1548

c. 50 Gujarati bigrams from Crubadan project resources

f125 {કે આ},377,10-06-2016-1210

f168 {જાય છે},571,09-06-2016-2035

f127 {જમે કે},39,10-06-2016-1210

f121 {થયો હતો},36,10-06-2016-1209

f145 {છે તે},465,09-06-2016-2037

f144 {શકાય છે},45,09-06-2016-2037

f161 {આવે છે},511,09-06-2016-1800

f162 {ના રોજ},53,09-06-2016-2044

f165 {કરવામાં આવી},56,09-06-2016-2044

f164 {એક જ},547,09-06-2016-2045

f166 {ખૂબ જ},56,10-06-2016-1210

f197 {છે જે},802,09-06-2016-2036

f109 {ખાસ કરીને},3,09-06-2016-2045

f108 {આવેલું છે},3,09-06-2016-2044

f107 {થાય છે},294,09-06-2016-1800

f147 {શકે છે},468,09-06-2016-2035

f101 {જોવા મળે},28,09-06-2016-2044

f43 {હતી અને},118,09-06-2016-2045

f184 {સામાન્ય રીતે},7,09-06-2016-2044

f183 {છે અને},696,09-06-2016-1800

f88 {કર્યું હતું},21,10-06-2016-1210

f122 {કરી શકાય},36,10-06-2016-1210

f26 {સમાવેશ થાય},1,09-06-2016-2036

f68 {જો કે},179,10-06-2016-1209

f64 {રહે છે},168,09-06-2016-2037

f63 {કારણ કે},166,09-06-2016-2045

f61 {કરવા માટે},160,09-06-2016-2037

f60 {કરવામાં આવે},16,09-06-2016-2035

f153 {પડે છે},498,09-06-2016-2044

f155 {ધરાવે છે},5,09-06-2016-2036

f177 {એ જ},654,09-06-2016-2044

f206 {કરી શકે},99,09-06-2016-2045

f128 {કરે છે},390,09-06-2016-1800

f90 {કર્યો હતો},22,09-06-2016-2045

f91 {મળે છે},225,09-06-2016-2035

f94 {આપે છે},264,09-06-2016-2044

f115 {કહે છે},306,09-06-2016-2037

f116 {આવી હતી},31,09-06-2016-2044

f131 {લાગે છે},399,09-06-2016-2045

f110 {આવ્યો હતો},3,10-06-2016-1210

f201 {હોય તો},899,09-06-2016-2037

f143 {અને આ},444,10-06-2016-1210

f55 {હોય છે},1402,09-06-2016-1800

f190 {કરી હતી},78,09-06-2016-2037

f58 {જ છે },1490,10-06-2016-1209

f30 {આવ્યું હતું},10,09-06-2016-2045

f139 {કહેવાય છે },43,09-06-2016-2045

f37 {છે કે},1098,09-06-2016-2035

f71 {અને તે},186,09-06-2016-2044

f72 {તેમ જ},19,09-06-2016-2045

2. Query details for geo-location based search (with 23 African cities)

Selection includes cities, which are capitals, commercial centers and those with largest population in their country.

Latitude	Longitude	Area	Tweets	Date-time	Location
1.078444	34.181006	500km	585	22-05-2016-1929	**mbale**
-6.130671	23.596658	500km	796	22-05-2016-1911	**mbujimayi**
-33.924868	18.424055	500km	993	07-06-2016-1517	**capetown**
-3.947926	29.623837	500km	1198	25-05-2016-1528	**bururi**
-6.162959	35.751607	500km	1194	26-05-2016-1753	**dodoma**
-18.87919	47.507905	500km	7895	07-06-2016-1507	**antananarivo**
-29.85868	31.02184	500km	7899	07-06-2016-1513	**durban**
-25.891968	32.605135	500km	9796	07-06-2016-1414	**maputo**
-6.165917	39.202641	500km	1386	26-05-2016-1755	**zanzibar**
-4.441931	15.266293	500km	891	22-05-2016-1823	**kinshasa**
-26.305448	31.136672	500km	3697	22-05-2016-1952	**mbabane**
-17.825166	31.03351	500km	10000	22-05-2016-1812	**harare**
-24.628208	25.923147	500km	10000	23-05-2016-1358	**gaborone**
-13.962612	33.774119	500km	10000	25-05-2016-1507	**lilongwe**
-11.687603	27.502617	500km	1098	07-06-2016-1405	**lubumbashi**
-4.26336	15.242885	500km	2077	22-05-2016-1821	**brazzaville**
-4.043477	39.668207	500km	198	07-06-2016-1520	**mombasa**
0.347596	32.58252	500km	2796	22-05-2016-1924	**kampala**
-29.363219	27.51436	500km	7793	26-05-2016-1814	**maseru**
-11.716734	43.368079	500km	2895	17-03-2017-2230	**Moroni,comoros**
-15.40669	28.28713	500km	1194	18-03-2017-0758	**Lusak,comoros**
-12.809645	45.130741	500km	1598	17-03-2017-1834	**mayotte**
-1.970579	30.104429	500km	197	25-05-2016-1518	**kigali**

Appendix 3 Details of ATEF components

1. DVM file contents

```
** DVM: morphological "variables" (attributes) for an AM phase aiming at
   handling indian tweets, potentially multilingual -- containing Hindi, Marathi,
Gujarati and English.

** For the moment, the language code is HIN (Hindi), but we may change it later to
ITW (Indian TWeets).

-EXC-   ** Exclusive variables.

** Language : EN-English, FR-French, HI-Hindi, MR-Marathi, GU-Gujarati,
              XML-xml tags, EMO-emojis.
LANG == (EN, FR, HI, MR, GU, XML, EMO).

** Tactical variable, the FINAL value indicates that we want to force a unique
result for AM.
TAKTIK  == ( FINAL
           ).

-NEX- ** Non-exclusive variables (set values).

** Dictionaries of bases (radicals) and affixes.
DICT == (1, 2, 3, 4, 5, 6). ** See later.
** By default,
   1: prefixes (all languages considered, but they will contain their language id)
   2: radicals (Hindi)
   3: suffixes (all languages considered, but they will contain their language id)
   4: radicals (English)
   5: radicals (Marathi)
   6: radicals (Gujarati).

** Typography (capitalization) of an occurence (indian tweets often contain English
words!).
TYPOG == ( ALLUPP,  ** ALL UPPercase.
           FIRSTUP, ** FIRST UPpercase.
           ABBREV,  ** ABBREViation ending with a period.
           SHILEFT  ** SHort I goes Left 1 1/2 consonant.
         ).
```

2. DVS file contents

```
** DVS: "syntactic" (actually syntactic, morphological or semantic!) variables"
(attributes) for an AM phase aiming at handling Indian tweets, potentially
multilingual -- containing Hindi, Marathi, Gujarati and English.

** Creation: 13/09/2016.
** Updated by Ritesh, 20/09/16: added categories for Number and Person
** Updated by Ritesh, 11/03/17: added 'ANY' to morphological gender
** Updated by Ritesh,12/03/17: added ASPECT category and moved 'Habitual' categ.
from MOOD to ASPECT
-EXC-
** Subcategory of Nouns.
SUBN  == (C,   ** Common.
          P,   ** Proper.
          V,   ** Verbal.
          ST   ** Spatio-temporal.
        ).

** Subcategory of Verbs.
SUBV  == (M,   ** Main.
          A    ** Auxiliary.
         ).
```

```
** Subcategory of Pronouns.
SUBP  == (PR,  ** PeRsonal.
          RF,  ** ReFlexive.
          RC,  ** ReCiprocal.
          RL,  ** ReLative.
          WH   ** WH-pronoun (interrogative).
         ).

** Subcategory of adJuncts of nouns.
SUBJ  == (ADJ, ** ADJective.
          Q    ** Quantifier.
         ).

** Subcategory of demonstrative.
SUBD  == (AB,  ** ABsolute -- that.
          RL,  ** ReLative -- Dravidian language ??? -- seems to be wrong.
          WH   ** WH-demonstrative -- Dravidian language ??? -- seems to be wrong.
         ).

** Subcategory of adverbs.
SUBA  == (MN,  ** MaNner.
          LC,  ** LoCation.
          TM,  ** TiMe -- added.
          DG   ** DeGree -- added.
         ).

** Subcategory of participles.
SUBL  == (RL,  ** adjectival (ReLational?).
          V,   ** adVerbial.
          N,   ** Nominal -- building.
          C    ** Conditional -- ???.
         ).

** Subcategory of postpositions: none.

** Subcategory of "particles".
SUBC  == (CD,  ** CoorDination.
          SB,  ** SuBordination conjunction.
          CL,  ** CLassifier.
          IN,  ** INterjection.
          X    ** others (phatics?)
         ).

** Subcategory of punctuations.
SUBPU == (SIPUL, ** SIngle PUnctuation Left: "itemizer" such as hyphen or dash
                    after a line break like \n or <br> or </h1>,
                    or "enumerator" such as 1-2-4-a) or ii).
          SIPUR, ** SIngle PUnctuation Right (period, comma, ellipsis);
                    colon, semi-colon, interrogation sign, exclamation sign).
          SIHT,  ** SIngle HTml tags -- monotags (ex: <img a-v list />, to be
                    preprocessed into for example %%HTMMONOTAG_img_24).
          DOPUL, ** DOuble PUnctuation Left  (opening quote, parenthesis, bracket,
                    brace, parenthetical dash).
          DOPUR, ** DOuble PUnctuation Right (closing quote, parenthesis, bracket,
                    brace, parenthetical dash).
          DOHTL, ** DOuble HTml tags Left  (<i a-v list>, to be preprocessed into
                    for example %%HTMOPENTAG_i_25).
          DOHTR, ** DOuble HTml tags Right (</i>, to be preprocessed into for
                    example %%HTMCLOSETAG_i_25).
          X      ** others (phatics?)
         ).

** Subcategory of emotional signs.
SUBEM == (EMOLIST, ** EMoji list.
          PHATIC,  ** Hmmm!, Uh!, Aha! etc.
         ).

** Subcategory of tweet-specific occurences.
```

```
SUBTW == (TWCD,  ** TWeet COmmand (such as RT for ReTweet).
          TWAD   ** TWeet ADdress (such as @Ritesh).
         ).

** Subcategory of Out of Text elements (hors-texte in French).
SUBOT == (IMAGE,    ** image or icon, can function as a proper singular noun (for
                       example, %%IMG_15).
          MATHEXP,  ** can function as a noun (for example, $ab+2$ preprocessed as
                       %%MEXP_13).
          MATHREL,  ** can function as noun or verbal kernel (for example, $xy>2$
                       preprocessed as %%MREL_43).
          PARTNAME, ** like BS-A-123-eBC.
          PRODNAME, ** like OS X El Capitan version 10-11-6.
          CHEMELEM, ** chemical element like H_2O or Al_2O_3, possibly preprocessed
                       as %%CHEMELEM_22.
          CHEMFORM, ** CHEMical FORMula (like benzines, preprocessed as
                       %%CHEMFORM_23.
          SYSSTR,   ** like menu names, menu items, command lines, system answers,
                       etc.
         ).

** Subcategory of residuals.
SUBRD == (F,   ** Foreign word).
          S,   ** Symbol (such as €, £, %).
          N,   ** Nominal -- building.
          UNK   ** Unknown.
        ).

-NEX-

** Morphosyntactic category (for terminals in a classical PSG).
CAT  == ( N,   ** Noun.
          V,   ** Verb.
          P,   ** Pronoun.
          J,   ** Nominal modifier (adJective).
          D,   ** Demonstrative.
          A,   ** Adverb.
          L,   ** participLe.
          PP,  ** PostPosition.
          C,   ** partiCle.
          PU,  ** PUnctuation.
          EM,  ** EMoji or phatic.
          OT,  ** Out of Text (hors-texte in French).
          TWCD,** TWeet COmmand (such as RT for ReTweet).
          RD   ** ResiDual.
        ).

** Morphological Mood of inflected verbs.
MOOD  == (DCL,   ** DeCLarative.
          SBJ,   ** SuBJunctive.
          CND,   ** CoNDitional.
          IMP,   ** IMPerative.
          PSM,   ** PreSuMptive.
          ABT    ** ABiliTative.
         ).
** Morphological Aspect of inflected verbs.
ASP  ==  (HAB,    ** HABitual.
         PRG,    ** PRoGressive.
         PFT,    ** PerFecTive.
          CML     ** CoMpLetive.
        ).
** Morphological Tense of inflected verbs.
TNS  ==  (PRS,   ** PReSent.
          PST,   ** PaST.
          FUT    ** FUTure.
         ).

** Morphological case.
```

APPENDIX 3

```
CAS  ==  (DIR,    ** DIRect.
          OBL     ** OBLique.
        ).

** Morphological gender.
GEN  ==  (MAS,    ** MASculine.
          FEM,    ** FEMinine.
          NEU,    ** NEUtral.
          ANY     ** ANY.
         ).

** Number.
NUM  ==  (SNG,    ** SiNGular.
          PLR,    ** PLuRal.
          ANY     ** ANY.
         ).

** Person.
PER  ==  (FT,     ** FirsT.
          SD,     ** SeconD.
          TD,     ** ThirD.
          SDHN ** SeconD person HoNorific.
         ).

-FIN-
```

Appendix 4 Indian language resources (MA, dictionaries)

1. Morphological analyses output from CFILT MA

Input : चौंका बेटीयों बेटियों प्रोफेसरों पूछिए

```
---------- Set of Roots and Features ----------
Token : चौंका,  Total Output : 2
[ Root : चौंका, Class : , Category : verb, Suffix : Null ]
        [ Gender : x, Number : x, Person : x, Case : x, Tense : x, Aspect : x, Mood : x ]
[ Root : चौंक, Class : , Category : verb, Suffix : ा ]
        [ Gender : +masc, Number : -pl, Person : x, Case : x, Tense : x, Aspect : +perfect, Mood : x ]
---------- End of Result ----------
---------- Set of Roots and Features ----------
Token : बेटीयों,  Total Output : 0
---------- End of Result ----------
---------- Set of Roots and Features ----------
Token : बेटियों,  Total Output : 1
[ Root : बेटी, Class : B, Category : noun, Suffix : यों ]
        [ Gender : -masc, Number : +pl, Person : x, Case : +oblique, Tense : x, Aspect : x, Mood : x ]
        [ Gender : -masc, Number : +pl, Person : x, Case : +oblique, Tense : x, Aspect : x, Mood : x ]
        [ Gender : -masc, Number : +pl, Person : x, Case : +oblique, Tense : x, Aspect : x, Mood : x ]
---------- End of Result ----------
---------- Set of Roots and Features ----------
Token : प्रोफेसरों,  Total Output : 0
---------- End of Result ----------
---------- Set of Roots and Features ----------
Token : पूछिए,  Total Output : 1
[ Root : पूछ, Class : , Category : verb, Suffix : ए ]
        [ Gender : +masc, Number : +pl, Person : x, Case : x, Tense : x, Aspect : +perfect, Mood : x ]
---------- End of Result ----------
```

2. Morphological analyses output from IIIT public web service[86]

Input: चौंका बेटीयों बेटियों प्रोफेसरों पूछिए

Address	TOKEN	Features (af='root,cat,gen,num,per,case,tam,suff')
1	चौंका	<fs af='चौंका,v,any,any,any,,0,0'>
		<fs af='चौंका,adj,m,sg,,d,,'>
		<fs af='चौंका,n,m,sg,3,d,0,0'>
		<fs af='चौंक,v,m,sg,any,,या,yA'>
2	बेटीयों	<fs af='बेटीयों,unk,,,,,,'>
3	बेटियों	<fs af='बेटी,n,f,pl,3,o,0,0'>
4	प्रोफेसरों	<fs af='प्रोफेसर,n,m,pl,3,o,0,0'>
5	पूछिए	<fs af='पूछ,v,any,sg,2,,ए,e' hon='y'>
		<fs af='पूछ,v,any,pl,2,,ए,e' hon='y'>

[86] http://sampark.iiit.ac.in/hindimorph/web/restapi.php/indic/morphclient

Appendix 5 Resource construction and normalisation

{Attribute → Value} mapping while constructing resource in Ariane-G5 format with source values from different columns in the LexDB

LexDB values	Transformed canonical form	LexDB values	Transformed canonical form
(schema attribute: root)		*(schema attribute: gender)*	
'-', 'x'	CAT(CAT0)	'-', 'x'	GEN(GEN0)
'n', 'noun'	CAT(N)	'm', '+masc'	GEN(MAS)
'proper noun'	CAT(N), SUBN(P)	'f', '-masc'	GEN(FEM)
'nst'	CAT(N), SUBN(ST)	'any', '+-masc'	GEN(MAS, FEM)
'v', 'verb'	CAT(V), SUBV(M)	*(schema attribute: number)*	
'verb_aux'	CAT(V), SUBV(A)	'-', 'x'	NUM(NUM0)
'sh_n', 'psp', 'post position'	CAT(PP)	'sg', '-pl'	NUM(SNG)
'adj', 'adjective'	CAT(J), SUBJ(ADJ)	'pl', '+pl'	NUM(PLR)
'quantifier	CAT(J), SUBJ(Q)	'any', '+-pl'	NUM(SNG, PLR)
'pn', 'pronoun'	CAT(P)	*(schema attribute: person)*	
'interrogative pronoun', 'interrogative'	CAT(P), SUBP(WH)	'-', 'x'	PER(PER0)
'punc'	CAT(PU)	'1', '1p', '(1p:2phon:3p)	PER(FT)
'adv'	CAT(A)	'2', '2p'	PER(SD)
'participle'	CAT(L)	'3', '3p', '(1p:2phon:3p)	PER(TD)
'demonstrative'	CAT(D)	'2h', '(1p:2phon:3p)	PER(SDHN)
'avy', 'particle'	CAT(C)	'any'	PER(FT,SD,TD,SDHN)
'conjunction'	CAT(C), SUBC(SB)	*(schema attribute: tense)*	
'interjection'	CAT(EM)	'-', 'x'	TNS(TNS0)
'unk'	CAT(RD)	'pr', '-past'	TNS(PRS)
(schema attribute: aspect)		'pa', '+past', '(+past'	TNS(PST)
'-', 'x'	ASP(ASP0)	'fut', '+future'	TNS(FUT)
'(-perfect:+habitual)', '+infinitive+habitual'	ASP(HAB)	*(schema attribute: mood)*	
'(-perfect : -habitual)'	ASP(PRG)	'-', 'x'	MOOD(MOOD0)
'(+perfect : +completive)', '(-perfect : -habitual)	ASP(CML)	'subjunctive','+conditional+subjunctive'	MOOD(SBJ)
'+infinitive', '+infinitive+habitual', '(-perfect:-habitual)	ASP(INF)	'+conditional','+imperative+conditional', '+ability+conditional', '+conditional+subjunctive'	MOOD(CND)
'+perfect' , '+perfect+subjunctive', '(+perfect:+completive)', '+perfect+ability','+perfect+inceptive', '+perfect+infinitive'	ASP(PFT)	'(+imperative:+polite)', '(+imperative:+intimate)','(+imperative : polite)','(+imperative:intimate)', '(+imperative',(+deontic:+obligation)', '(+deontic:+necessity)', '+imperative+conditional'	MOOD(IMP)
(schema attribute: case)		'(+probability : +ability)', '+perfect+ability','+ability', '+ability+conditional'	MOOD(ABT)
'-', 'x'	CAS(CAS0)	'dcl'	MOOD(DCL)
'd', '-obl', '-oblique'	CAS(DIR)	'psm'	MOOD(PSM)
'o', '+obl', '+oblique'	CAS(OBL)		
'any', '+-obl'	CAS(DIR,OBL)		

Appendix 6 Tweets non-understandable in source language

Examples of tweets not understandable in Hindi

1.	और भादो के?
2.	#Sarahah @sarahah_com DM करो
3.	#दोगली_सरकार सहीक्या 100% सही कहाहै भृष्टमीडिया व भृष्ट सरकार #संतरामपालजीनिर्दोष को साजिसके तहतअरेस्टकिया इधरअश्लीलतावाले संतकेआगे नतमस्तक
4.	Replying to @yadavtejashwi शाहबुद्दीन ने किस गांधी के रास्ते पे चल के कांड कावड़िया था???
5.	घीया त्यौहार की बहुत-सी बधाई, खूब घी खाया, यो दिन यो मास भेटणै रया झुमेंल चौरासी जाण बै ह गे उमैल भादो औण लैरे। यो ऋतु मास भेंटणै रया,जी राय
6.	#जीने_की_राह भगती भाव भादो नदी सभी चले गहराइसरिता सोई जानिए जेठ मास टहराय रहस्य
7.	#जीने_की_राह भगती भाव भादो नदी सभी चले गहराइसरिता सोई जानिए जेठ मास टहराय रहस्य

Examples of tweets not understandable in Japanese

1.	@muramumumumu うにゅ?。みて欲しいにゃよ?。今日はインするのかにゃ？それよりも、みんなちゃんと帰れるのかにゃ?、雪が大変そうにゃょ。。。
2.	チョコレイト･ディスコ 2014\n 焼きたて持って、雪の中映画館へ。 http://t.co/hnsTJdl0Fu

Appendix 7 Morphological analyses of code-mixed tweets by ATEF

1. गोवा के मंत्री ने गैंगरेप को फिर बताया 'छोटी घटना' </tweet>

1 '': UL('ULTXT')
2 '': UL('ULFRA')
3 '': UL('ULOCC')
4 'गोवा': UL('गोवा'), LANG (HI), CAT(N), CAS(DIR,OBL), GEN(MAS), NUM(SNG,PLR), PER(TD)
5 '': UL('ULOCC')
6 'के': UL('का'), LANG (HI), CAT(PP), CAS(DIR,OBL), GEN(MAS), NUM(SNG,PLR)
7 '': UL('ULOCC')
8 'मंत्री': UL('मंत्री'), CAT(RD)
9 '': UL('ULOCC')
10 'ने': UL('ने'), CAT(RD)
11 '': UL('ULOCC')
12 'गैंगरेप': UL('गैंगरेप'), CAT(RD)
13 '': UL('ULOCC')
14 'को': UL('को'), LANG (HI), CAT(PP)
15 '': UL('ULOCC')
16 'फिर': UL('फिर'), CAT(RD)
17 '': UL('ULOCC')
18 'बताया': UL('बताया'), CAT(RD)
19 '': UL('ULOCC')
20 "छोटी': UL('_छोटी'), CAT(RD)
21 '': UL('ULOCC')
22 'घटना": UL('घटना_'), CAT(RD)
23 '': UL('ULOCC')
24 '<tweet/>': UL('</tweet>'), LANG (XML)

2. RT @viveklkw: #बुरा_ना_मानो_होली_है ⃠ @arvindkejriwal का हाल-ए-पंजाब-गोवा : फटी पड़ी है कलेजा दबाये बैठे है .. गए थे मारने और खुद मरवाए… <tweet/>

4337 '': UL('ULFRA')
4338 '': UL('ULOCC')
4339 'RT': UL('RT'), CAT(RD)
4340 '': UL('ULOCC')
4341 '@viveklkw:': UL('@viveklkw:'), CAT(RD)
4342 '': UL('ULOCC')
4343 '#बुरा_ना_मानो_होली_है': UL('#बुरा_ना_मानो_होली_है'), CAT(RD)

4344 '': UL('ULOCC')

4345 '⃠': UL('⃠'), CAT(RD)

4346 '': UL('ULOCC')

4347 '@arvindkejriwal': UL('@arvindkejriwal'), CAT(RD)

4348 '': UL('ULOCC')

4349 'का': UL('का'), LANG (HI), CAT(PP), CAS(DIR), GEN(MAS), NUM(SNG)

4350 '': UL('ULOCC')

4351 'हाल-ए-पंजाब-गोवा': UL('हाल-ए-पंजाब-गोवा'), CAT(RD)

4352 '': UL('ULOCC')

4353 ':': UL(':'), CAT(RD)

4354 '': UL('ULOCC')

4355 'फटी': UL('फटी'), CAT(RD)

4356 '': UL('ULOCC')

4357 'पड़ी': UL('पड़ी'), CAT(RD)

4358 '': UL('ULOCC')

4359 'है': UL('है'), CAT(RD)

4360 '': UL('ULOCC')

4361 'कलेजा': UL('कलेजा'), CAT(RD)

4362 '': UL('ULOCC')

4363 'दबाये': UL('दबाये'), CAT(RD)

4364 '': UL('ULOCC')

4365 'बैठे': UL('बैठे'), CAT(RD)

4366 '': UL('ULOCC')

4367 'है': UL('है'), CAT(RD)

4368 '': UL('ULOCC')

4369 '..': UL('..'), CAT(RD)

4370 '': UL('ULOCC')

4371 'गए': UL('जा'), LANG (HI), CAT(V), ASP(PFT), GEN(MAS), NUM(PLR), PER(FT,SD,TD,SDHN), SUBV(M)

4372 '': UL('ULOCC')

4373 'थे': UL('थे'), CAT(RD)

4374 '': UL('ULOCC')

4375 'मारने': UL('मारने'), CAT(RD)

4376 '': UL('ULOCC')

4377 'और': UL('और'), LANG (HI), SUBC (SB), CAT(N,J,A,C), CAS(DIR,OBL), GEN(MAS,FEM), NUM(SNG,PLR), PER(TD), SUBJ(Q)

4378 '': UL('ULOCC')

4379 'खुद': UL('खुद'), LANG (HI), CAT(V,A), GEN(MAS,FEM), NUM(SNG,PLR), PER(FT,SD,TD,SDHN), SUBV(M)

4380 '': UL('ULOCC')

4381 'मरवाए...': UL('मरवाए...'), CAT(RD)

4382 '': UL('ULOCC')

4383 '<tweet/>': UL('</tweet>'), LANG (XML)

Abstract

As TWITTER evolves into a ubiquitous information dissemination tool, understanding tweets in foreign languages becomes an important and difficult problem. Because of the inherent code-mixed[87], disfluent and noisy nature of tweets, state-of-the-art Machine Translation (MT) is not a viable option (Farzindar & Inkpen, 2015). Indeed, at least for Hindi and Japanese, we observe that the percentage of "understandable" tweets falls from 80% for natives to below 30% for target (English or French) readers using GOOGLE TRANSLATE or YANDEX. Our starting hypothesis is that it should be possible to build generic tools, which would enable foreigners to make sense of at least 70% of "native tweets", using a versatile "active reading" (AR) interface, while simultaneously determining the percentage of understandable tweets under which such a system would be deemed useless by intended users.

We have thus specified a generic "SUFT" (System for helping Understand Foreign Tweets), and implemented SUFT-1, an interactive multi-layout system based on AR, and easily configurable by adding dictionaries, morphological modules, and MT plugins. It is capable of accessing multiple dictionaries for each source language and provides an evaluation interface. For evaluations, we introduce a task-related measure inducing a negligible cost, and a methodology aimed at enabling a « continuous evaluation on open data », as opposed to classical measures based on test sets related to closed learning sets. We propose to combine *understandability ratio* and *understandability decision time* as a two-pronged quality measure, one subjective and the other objective, and experimentally ascertain that a dictionary-based active reading presentation can indeed help understand tweets better than available MT systems.

In addition to gathering various lexical resources, we constructed a large resource of "word forms" appearing in Indian tweets with their morphological analyses (163221 Hindi word forms from 68788 lemmas and 72312 Marathi word forms from 6026 lemmas) for creating a multilingual morphological analyzer specialized to tweets, which can handle code-mixed tweets, compute unified features, and present a tweet with an attached AR graph from which foreign readers can intuitively extract a plausible meaning, if any.

Résumé

Alors que TWITTER évolue vers un outil omniprésent de diffusion de l'information, la compréhension des tweets en langues étrangères devient un problème important et difficile. En raison de la nature intrinsèquement à commutation de code, discrète et bruitée des tweets, la traduction automatique (MT) à l'état de l'art n'est pas une option viable (Farzindar & Inkpen, 2015). En effet, au moins pour le hindi et le japonais, nous observons que le pourcentage de tweets « compréhensibles » passe de 80% pour les locuteurs natifs à moins de 30% pour les lecteurs en langue cible (anglais ou français) utilisant GOOGLE TRANSLATE ou YANDEX. Notre hypothèse de départ est qu'il devrait être possible de créer des outils génériques, permettant aux étrangers de comprendre au moins 70% des « tweets locaux », en utilisant une interface polyvalente de « lecture active » (LA, AR en anglais) tout en déterminant simultanément le pourcentage de tweets compréhensibles en-dessous duquel un tel système serait jugé inutile par les utilisateurs prévus.

Nous avons donc spécifié un « SUFT » (système d'aide à la compréhension des tweets étrangers) générique, et mis en œuvre SUFT-1, un système interactif à présentation multiple basé sur la LA, et facilement configurable en ajoutant des dictionnaires, des modules morphologiques et des plugins de TA. Il est capable d'accéder à plusieurs dictionnaires pour chaque langue source et fournit une interface d'évaluation. Pour les évaluations, nous introduisons une mesure liée à la tâche induisant un coût négligeable, et une méthodologie visant à permettre une « évaluation continue sur des données ouvertes », par opposition aux mesures classiques basées sur des jeux de test liés à des ensembles d'apprentissage fermés. Nous proposons de combiner le *taux de compréhensibilité* et le *temps de décision de compréhensibilité* comme une mesure de qualité à deux volets, subjectif et objectif, et de vérifier expérimentalement qu'une présentation de type lecture active, basée sur un dictionnaire, peut effectivement aider à comprendre les tweets mieux que les systèmes de TA disponibles.

En plus de rassembler diverses ressources lexicales, nous avons construit une grande ressource de "formes de mots" apparaissant dans les tweets indiens, avec leurs analyses morphologiques (163221 formes de mots hindi dérivées de 68788 lemmes et 72312 formes de mots marathi dérivées de 6026 lemmes) pour créer un analyseur morphologique multilingue spécialisé pour les tweets, capable de gérer des tweets à commutation de code, de calculer des traits unifiés, et de présenter un tweet en lui attachant un graphe de LA à partir duquel des lecteurs étrangers peuvent extraire intuitivement une signification plausible, s'il y en a une.

[87] See Definition 3.