Tense Conversion in NLP with focus on finding the verb and obtaining its root form

Abdul Raheem Usmani,
Chayan Pandey,
Nupur Agrawal ,Prakriti Gautam

Department of Information
Technoogy,
Galgotias College of Engg. And Technology
Greater Noida

ABSTRACT
Tense Conversion is a sub requirement in a large number ofapplications that generate natural language,examples may be document summarisers, chatterbots etc.While the popular techniques for tense conversion would be to parse the entire sentence,this paper proposes a technique to reduce the amoune of tokens to be analysed for tense conversionby eliminating the tokens that remain unaffected by tense conversion.

1.0 INTRODUCTION

To convert an English language sentence from one form to another the core effort lies in identifying the verb. Once the verb has been identified, it is analysed for it is analysed for its current tense form and the proper transformation rule are applied to it to obtain the sentence in the desired tense.

The Traditional parsing techniques are usually deployed to obtain the POS tag for each token in the sentence.The verb can thus be obtained and modified accordingly. POS tagging involves complete semantic and syntactic analusis of all the tokens generated from the sentence. This requires perfroming a search on a large database that represents the knowledge of the system for all the tokens in the sentence.All this precedes the syntactic analysis of the sentence. Thus large amount of work of work needs to be done which may not be required.

2.0 REDUCTION IN THE AMOUNT OF DATA TO BE SEARCHED

In order to reduce the amount of data to be lookedup in the knowledge base of the system, we propose to use some commonly known properties of English language sentences. We first propose to perform a search on smaller databases that contains certain special category of words that do not undergo any change during the tense transformation of the sentence. Further, these words also would tell about the words preceeding or succeeding them, so that they may be further categorized accordingly.

Some of these categories of words are:

Determiners

Determiners are a kind of noun modifier; they precede and are necessarily followed by nouns. While adjectives perform a similar function, the term 'determiner' refers to a relatively limited set of well-established words that can be said to 'mark' nouns.[2]

*The function of determiners is to 'express reference'; i.e. they clarify what a noun is referring to. For e.g. when one says '*that *box', the listener knows which box is being referred to.*

Determiners themselves do not undergo any change during tense conversion . Also a determiner ionforms ud that the word following it is a a noun or an adjective which also do not undergo any change during tense transformation.

There are around 150 determiners inEnglish language.

Prepositions

A word that shows the relationship between a noun or pronoun and other words in a sentence.

ISBN: 9788175157538

A prepostion governs, and usually precedes, a noun or pronoun and expressing a relation to another word or element in the clause, as in... as in “the man *on* the platform,” “she arrived *after* dinner,”

There are about 150 prepositions in English.[3]

Conjunctions

Conjunctions are words that join two phrases or sentences

Conjunctions also do not undergo any change during tense transformation of sentences.

Thus the tokens that represent either of these categories must not be searched in the knowledge base, also the information they gives about the word preceding or succeeding them must be utilised to further reduce the search.

The rest of the tokens contains verb along with only few non verb tokens , they must then be processed to obtain the root verb.

3.0 DETERMINING THE ROOT VERB

The primary step in our process is the determination of the base verb. Doing so requires anlysing the word according to the following table

Table 1: Patterns of Participles and Tenses of Verbs [4]

Key:	=>	becomes
	!	not (EXCEPTION to the rule)
	==	equals
	!=	not equals

Classifications (ends with)	Sub-Classes (ends with)	Variants (ends with)	Action	Examples
S *PRESENT PARTICIPLE*	ies	== dies, ties	Remove last character	Ties => tie
		All others	replace last 3 characters with ‘y’	Carries => carry
	us	none	No action	focus
	es	Vowel + consonant + es	Remove last character	Scores = score
		others	Remove last 2 characters	
It *PAST TENSE*	it	== bit	Add ‘e’	Bit => bite
Ought *PAST TENSE/ PAST PARTICIPLE*	thought		Replace last 5 characters with ‘ink’	thought => think
	fought		Replace last 5 characters with ‘ight’	Fought => fight
	sought		Replace last 5 characters with ‘eek’	Sought => seek
	bought		Replace last 5 characters with ‘uy’	Bought => buy
	brought		Replace last 5 characters with	Brought =>

			‘ing’	bring
Ang *PAST TENSE/ PAST PARTICIPLE*	Consonant + Sang, Consonant + rang, Consonant + tang, Consonant + wang		Replace last 3 characters with ‘ing’	Sang => sing
Aught *PAST TENSE/ PAST PARTICIPLE*	caught		Replace last 3 characters with ‘tch’	Caught => catch
	taught		Replace last 4 characters with ‘each’	Taught => teach
Wn *PAST PARTICIPLE*	R +Vowel + wn, s +Vowel + wn, h +Vowel + wn, n +Vowel + wn, l +Vowel + wn	!=Drown, !=clown, !=crown, != disown, != frown	Remove last character.	Grown => grow
Ew *PAST TENSE*	Blew		Replace last 2 characters with ‘ow’	Blew => blow
	flew		Replace last 2 characters with ‘y’	Flew => fly
	drew		Replace last 2 characters with ‘aw’	Drew => draw
Ept *PAST TENSE/ PAST PARTICIPLE*		!=accept	Replace last 3 characters with ‘eep’	Kept => keep
Ting *PRESENT PARICIPLE*	Iting, ating, outing, uoting		Replace last 3 characters with ‘e’	Uniting => unite
	eating	!= eating	Replace last 3 characters with ‘e’	Creating => create
	others		Remove last 3 characters	Voting => vote
ning (!ening) *PRESENT PARTICIPLE*	nning		Remove last 4 characters	Running => run
	uning, oning, ining, caning		Replace last 3 characters with ‘e’	Tuning => tune
	others		Remove last 3 characters	Burning => burn

Ing *PRESENT PARTICIPLE*	aking	eaking	Remove last 3 characters	Speaking => speak
		others	Replace last 3 characters with 'e'	Shaking => shake
	Vowel + consonant + ing		Replace last 3 characters with 'e'	Riding => ride
	lving, dging, gling, tling, ching, nging, bling, kling		Replace last 3 characters with 'e'	Sprinkling => sprinkle
	others		Remove last 3 characters	Hearing => hear
D *PAST TENSE/ PAST PARTICIPLE*	!dd, !rd, ! ld, !nd, !vowel + d		Remove last character	Heard => hear
Ed *PAST TENSE*	Gned, yed, ned, hed	nned	Remove last 3 characters	Banned => ban
		Consonant + Vowel + ned	Remove last character	hydroplaned => hydroplane
		Vowel + Vowel + ned	Remove last 2 characters	Bemeaned => bemean
		ched	Remove last character	Psyched => psych
	Led, bed	lled	Remove last 2 characters	Swelled => swell
		bbed	Remove last 3 characters	Stubbed => stub
		Consonant + Vowel + "sub-class"	Remove last character	Prescribed => prescribe
		Vowel + Vowel + "sub-class"	Remove last 2 characters	Pooled => pool
	Cked, rked, ssed		Remove last 2 characters	Passed => pass
	rred	Vowel + rred	Remove last 3 characters	Inferred => infer
	Med	Vowel + med	Remove last character	Timed => time

ISBN: 9788175157538

		mmed	Remove last 3 characters	Crammed => cram
	ured		Remove last character	Cured => cure
	ied	!died	Replace last 3 characters with ‘y’	Unified => unify
	red	Ared, ered, ired, ored	Remove last character	Stored => store
		Uired, tred	Remove last character	Acquired => acquire
	Tted, dded	!added	Remove last 3 characters	Batted => bat
	Vowel + ted	Oated, ooted, eeted, ieted, eited	Remove last 2 characters	Footed => foot
		dited	Remove last 2 characters	Edited => edit
		others	Remove last character	Violated => violate
	Ded, ved, ged, sed, ked, zed, wed, !=wed	Vowel + gged	Remove last 3 characters	Drugged => drug
		lked	Remove last 2 characters	Talked => talk
		Vowel + wed	Remove last 2 characters	Gnawed => gnaw
		others	Remove last character	Smoked => smoke
Id *PAST PARTICIPLE*	Aid, !=aid		Replace last 2 characters with ‘y’	Laid => lay
De *PAST TENSE/ PAST PARTICIPLE*	made		Replace last 2 characters with ‘ke’	Made => make
	bade		Replace last 3 characters with ‘id’	Bade => bid

Usage of the patterns is based on the following algorithm, represented in first-order logic, where an input verb (represented by *a*) is compared to the Classification (represented by *x*) and to the Sub-classes (represented by *y*) and to the variant (represented by *z*).

We make the following assumptions. First, that there is at least one verb, *a*, in the English language where pattern *x*
occurs.

$$a\ (verb(a) \rightarrow patternOccurs(a,x)) \quad (1)$$

Second, that there is at least one verb, *a*, in the English language where both pattern *x* and pattern *y* occur.

$$a\ (verb(a) \rightarrow patternOccurs\ (a,x) \quad patternOccurs\ (a,y)) \quad (2)$$

Third, that there is at least one verb, *a*, in the English language where pattern *x* and pattern *y* and pattern *z* occur.

$$a\ (verb(a) \rightarrow patternOccurs\ (a,x) \quad patternOccurs\ (a,y) \quad patternOccurs\ (a,z)) \quad (3)$$

Fourth, for all *z*, if the *z* is not specified (blank entry in Table 1) and no other corresponding *z* matched, then *z* is considered to occur in *a*.

$$z1,2,3...n \quad ((\neg patternOccurs\ (a,z1,2,3...n\text{-}1) \quad \neg specified\ (z_n)) \rightarrow patternOccurs\ (a,z)$$

Fifth, that for all *a* if pattern *x* and pattern *y* and pattern *z* occur, then *e* will not
occur.

$$a\ ((\ patternOccurs\ (a,x) \quad patternOccurs\ (a,y) \quad patternOccurs\ (a,z)) \leftrightarrow \neg patternOccurs(a,e)) \quad (5)$$

Therefore, if *a* falls into a pattern (*x*, *y*, *z*), then the corresponding action (represented by *b*) is taken if and only if any EXCEPTION to the rule (represented by *e*) does not occur.

$$changeVerb\ (a,\ x,\ y,\ z,\ b,\ e) =$$

$$patternOccurs\ (a,x) \quad patternOccurs\ (a,y) \quad (patternOccurs\ (a,z) \quad (\neg patternOccurs\ (a,z) \quad \neg specified\ (z)))$$

$$\neg patternOccurs(a,e)$$

Table 2 shows a random sampling of the complete results of the simulations.

Table 2: Random Test on Base Verb Generating Algorithm

Random Test 1 (Test tense/participle => generated Base Verb)	Random Test 2 (Test tense/participle => generated Base Verb)
allying => ally	accusing => accuse
anchylosed => anchylose	aromatizing => aromatize
averaged => average	autotomising => autotomise
backsplicing => backsplice	brabbled => brabble
brutalizing => brutalize	canoed => canoe
carnifying => carnify	caravanning => caravan
ceased => cease	cold-chiselling => cold-chisell
chroming => chrome	curing => cure
confiscated => confiscate	dabbled => dabble deoxidised
crapping => crap	=> deoxidise diphthongizing =>
denunciated => denunciate	diphthongize disprizing =>
ensured => ensure	disprize
evolving => evolve gnawn	divinized => divinize
=> gnaw halogenated =>	elegized => elegize
halogenate hoeing => hoe	encapsulating => encapsulate
installing => install	enthroned => enthrone
jargonizing => jargonize	flared => flare
jogging => jog	frivolled => frivoll
meditating => meditate	ideating => ideate
overcomplicated => overcomplicate	illiberalizing => illiberalize
pastoralizing => pastoralize	inosculated => inosculate
photoengraved => photoengrave	marshalled => marshall
preimitated => preimitate	outplodding => outplod
redisputed => redispute	outvoicing => outvoice
relosing => relose	overcultivated => overcultivate
remortgaging => remortgage	overidentified => overidentify
retraversing => retraverse	preadvertised => preadvertise
tenderizing => tenderize	prepledged => prepledge
terminated => terminate	prequarantining => prequarantine
underpopulating => underpopulate	prerefining => prerefine
upswept => upsweep	quasi-admiring => quasi-admire
vinylated => vinylate	quoting => quote
	recompensed => recompense

warbling => warble	reinduced => reinduce
	reutilized => reutilize
	sanitized => sanitize
	skywrote => skywrite
	surcharged => surcharge
	unfenced => unfence
	upttore => upttear
	vocalized => vocalize
	zigzagged => zigzag

4.0 Transformation rules

The following transformation rules can be used to transform the sentence once the root /base verb has been obtained:

Simple Present:
Subject+ V1(verb first form)+Object

Present Continous:
Subject+HV(Helping Verb)+v1+ing +object

Present Perfect:
Subject+ HV(has/have) +v3+object

Present Perfect Continous:
Subject+ HV(has/have) +been +v1+ing +object

Simple Past:
Subject+ V2+Object

Past Continous:
Subject+HV(was/ were)+v1+ing +object

Past Perfect:
Subject+ HV(had) +v3+object

Past Perfect Continous:
Subject+ HV(had) +been +v1+ing +object

Simple Futuret:
Subject+ HV(will/shall)+V1(verb first form)+Object

Future Continous:
Subject+HV(will/shall)+be+v1+ing +object

future Perfect:
Subject+will/shall+ HV(have) +v3+object

Future Perfect Continous:
Subject+ HV (will/shall)+have+been +v1+ing +object

Discussion and Conclusion :

Although the proposed algorithm handles easily the simple sentences, it does not give accurate results in sentences involving words that can both act as nouns and verbs. Also the complex and composite sentences' conversion accuracy needs to improve.

REFERENCES

[1] NLP and Information Retrieval,Tanveer Siddiqui,Oxford Learning Publications

[2] http://www.englishleap.com/grammar/determiners

[3] http://www.englishclub.com/grammar/prepositions-list.htm

[4] " A Pattern Based Approach for the Derivation of Base Forms of Verbs from Participles and Tenses for Flexible NLP. " ***Ram Gopal Raj and S. Abdul-Kareem*** ,Department of Artificial Intelligence
Faculty of Computer Science and Information Technology ,University of Malaya, Kuala Lumpur