키워드 추출 알고리즘을 이용한 이용 약관 요약 방법 설계

이규은*, 김상원**, 김기천*
*건국대학교 컴퓨터공학과
**건국대학교 컴퓨터정보통신공학과
e-mail : kelee0812@konkuk.ac.kr, llllssss94@konkuk.ac.kr, kckim@konkuk.ac.kr

Designing Method for Summarization of the Terms & Conditions using Keyword Extraction Algorithm

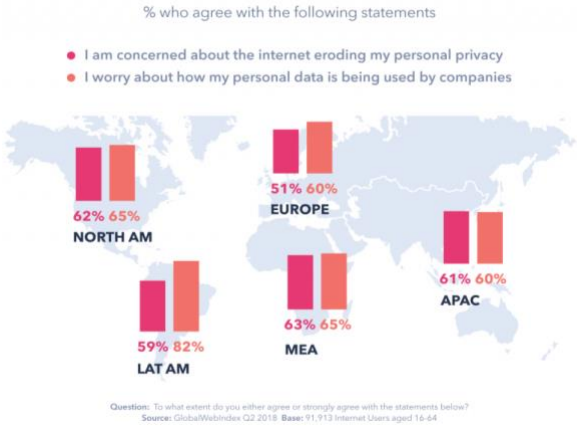
Kyu-Eun Lee*, Sang-Won Kim**, Kee-Cheon Kim*
*Department of Computer Science and Engineering, Konkuk University
** Department of Computer, Information & Communications Engineering, Konkuk University

요 약

소비자 개개인의 취향에 부합하는 맞춤형 서비스가 기업들에게 중요한 과제로 대두되고 있다. 이 같은 기대에 부응하기 위해 기업은 막대한 양의 데이터를 수집하고 그를 기반으로 서비스를 제공하려 노력하고 있다. 하지만, 데이터 수집 과정에서 어떤 정보가 어떻게 수집되는지를 명확히 아는 소비자는 극히 드물다. 소비자가 원하는 정보는 이용 약관에 명시되어 있지만, 정작 약관을 꼼꼼히 읽어보고 그에 동의하는 경우는 그리 많지 않기 때문이다. 이 때문에 소비자들이 원치 않는 정보가 수집되어도 이의를 제기하기 힘든 사태가 발생하기도 한다. 본 논문에서는 몇 가지 자연어 처리 기술을 이용하여 키워드를 추출해 약관의 핵심적인 내용을 요약 정리하는 알고리즘을 제안한다. 이를 통해 약관의 핵심 요약을 제공하여 이러한 소비자의 불이익을 줄일 수 있을 것으로 기대한다.

1. 서론

데이터는 현대 사회에서 가장 중요한 자원 중 하나이다. 과거에는 서양 열강들이 천연자원과 노동력을 찾아 식민지를 개발하고 다녔다면, 현재는 전 세계 대부분의 기업이 마르지 않는 자원, 데이터를 수집하기 위해 노력 중이다. 하지만, 그에 따라 개인정보에 대한 사람들의 염려도 커지고 있다.



(그림 1) 개인 정보에 대한 인식 설문 자료 [1]

2018년 'GlobalWebIndex'에서 91,913명의 16-64세 인터넷 사용자들을 대상으로 조사한 결과인 (그림 1)에 따르면, 다수의 응답자가 인터넷 상에서 개인정보를 침해당하는 것에 대해 염려하고 있으며, 기업에서 정보를 어떻게 이용하는가에 대해 불안감을 느끼고 있었다. 또한, 영국과 미국 응답자 중 72% 회사의 개인정보 수집에 대해 과거보다 관심이 늘었다고 답했으며, 개인정보를 존중하지 않는 기업들은 신뢰를 잃을 것이라고 전문가는 예상했다.[1] 하지만, 기업들이 개인정보를 무차별적으로 수집하고, 이용하는 사례는 끊이지 않는다. 일례로 최근 국내에서는 AI 스피커를 통해 수집한 음성데이터를 문자화하여 보관한 사실이 드러났다.[2] 그러나 사용자들의 불만에 그들은 약관에 명시가 되어있으니 큰 문제가 되지 않는다는 식으로 일관하였다. 우려스러운 점은 이용자의 대부분이 내용이 너무 길고 복잡하다는 이유로 개인정보 수집약관을 정독하지 않고 동의를 하곤 한다는 것이다.[3] 약관에 동의한 경우, 정보에 대한 권리를 주장하기 힘들뿐더러, 원치 않는 목적으로 이용되었더라도 문제를 제기하기 모호한 경우가 많다. 이러한 문제를 인지하고 본 논문에서는 약관 속의 민감할 수 있는 내용을 키워드로 지정하여 간단하고 명확하게 이용자에게 제시할 수 있는 방법을 제안한다.

2. 관련 연구

2.1 POS(Part-Of-Speech) tagging

자연어를 처리하는 과정에서 가장 먼저 해야 할 것은 문장의 요소들을 공백 또는 문장부호에 따라 쪼개는 tokenization 이다. 이 때, 가장 먼저 해야 할 일은 각각의 요소들을 token 으로 명시하는 것이다. 그 다음, 각 token 의 품사를 추정하고, 접미사를 확인해 복수형, 과거형 등으로 형태가 변화하였는지 파악해 원형을 찾는 lemmatization 을 진행한다. 이 과정은 해당 token 의 의미는 변화시키지 않기 때문에 원하는 keyword 를 추출하는 데에 도움이 될 뿐 아니라, 원형에서 어떤 형태로 변화하였는지를 알 수 있다. 게다가, token 이 해당 문장에서 어떤 역할을 수행하는지, 즉 문법적 특성까지 알 수 있다. 그 후, 문법적 특성을 단어에 부여하는 POS tagging 을 진행하면, 특성은 곧 그 token 의 tag 가 된다.

Table 2
The Penn Treebank POS tagset.

1.	CC	Coordinating conjunction	25.	TO	*to*
2.	CD	Cardinal number	26.	UH	Interjection
3.	DT	Determiner	27.	VB	Verb, base form
4.	EX	Existential *there*	28.	VBD	Verb, past tense
5.	FW	Foreign word	29.	VBG	Verb, gerund/present participle
6.	IN	Preposition/subordinating conjunction	30.	VBN	Verb, past participle
7.	JJ	Adjective	31.	VBP	Verb, non-3rd ps. sing. present
8.	JJR	Adjective, comparative	32.	VBZ	Verb, 3rd ps. sing. present
9.	JJS	Adjective, superlative	33.	WDT	*wh*-determiner
10.	LS	List item marker	34.	WP	*wh*-pronoun
11.	MD	Modal	35.	WP$	Possessive *wh*-pronoun
12.	NN	Noun, singular or mass	36.	WRB	*wh*-adverb
13.	NNS	Noun, plural	37.	#	Pound sign
14.	NNP	Proper noun, singular	38.	$	Dollar sign
15.	NNPS	Proper noun, plural	39.	.	Sentence-final punctuation
16.	PDT	Predeterminer	40.	,	Comma
17.	POS	Possessive ending	41.	:	Colon, semi-colon
18.	PRP	Personal pronoun	42.	(	Left bracket character
19.	PP$	Possessive pronoun	43.	)	Right bracket character
20.	RB	Adverb	44.	"	Straight double quote
21.	RBR	Adverb, comparative	45.	‘	Left open single quote
22.	RBS	Adverb, superlative	46.	“	Left open double quote
23.	RP	Particle	47.	’	Right close single quote
24.	SYM	Symbol (mathematical or scientific)	48.	”	Right close double quote

(그림 2) penn treeback POS tagset [4]

이러한 tag 들의 모음을 tagset 이라고 하는데, (그림 2)는 POS tagset 중 하나인 penn treeback POS tagset 를 나타낸다.[4] tagset 에 따라 tagging 하는 과정에서 각 token 은 명사, 동사 등의 품사로 tagging 될 수도 있고, 마침표나 쉼표, 수사같은 stop word 로 tagging 될 수도 있다.

2.2 noun chunking

Noun chunking 을 통해 각 token 의 연관관계를 파악해 하나로 묶는 과정을 거치고 나면, 각 token 이 의미하는 바가 분명해진다. Noun chunking 은 다음과 같이 동작한다. 먼저, tag 가 noun 이면서 relation 이 modifier, 즉 수식어가 아닌 token 을 chunk 의 root text 로 설정한 뒤, 해당 token 의 주변부에서 relation 이 modifier 인 token 들과 root token 으로 하나의 chunk 를 구성하여 root text 의 children 으로 한다. noun 이 아닌 token 들은 각 token 자체가 하나의 chunk 를 구성하게 된다.

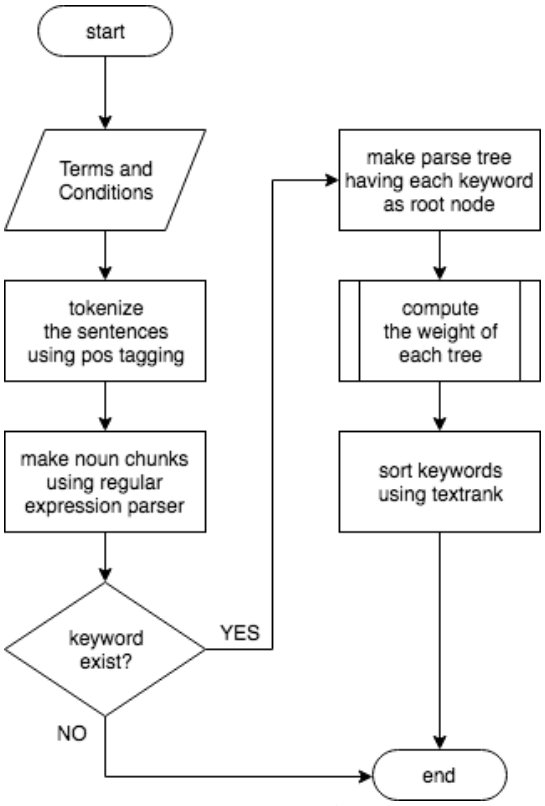
2.3 navigating parse tree

tokenization 과정이 끝난 후 각 token 들을 살펴보면, 문장에 단 하나만 존재하는 핵심 단어인 동사의 tag 는 ROOT 로 되어있다. 이 동사를 ROOT 로 하고, noun chunking 과정에서 만들어진 각 chunk 들의 children 을 leaf 단계까지 계속하여 연결하면 문장의 전체적인 계층 구조를 파악할 수 있는 parse tree 가 완성된다.

2.4 textrank

textrank 는 pagerank 에서 파생된 알고리즘이다. pagerank 는 웹페이지를 node 로 하는 그래프를 생성하고, 각 node 의 가중치를 계산해 그 결과값을 기반으로 순위를 결정하는 알고리즘이다. 이는 구글 등 검색 엔진에서 검색어에 맞춰 페이지 노출 순위를 결정하는 과정에 흔히 쓰인다. textrank 는 pagerank 의 페이지 대신 문장 속의 token 을 node 로 하고, 각 문장들의 가중치를 통해 우선 순위를 부여한다. 이 그래프에서 edge 는 방향성을 가지며, 각 node 의 가중치는 해당 node 와 연결된 inbound link, outbound link 의 가중치를 기반으로 연산해 도출한다.[5]

3. 제안 방법



(그림 3) 순서도

본 논문에서는 위에서 언급한 알고리즘을 기반으로 (그림 3)의 순서에 따라 키워드를 추출하고 결과를 도출해낸다.

예시로서, 네이버의 개인 정보 이용 약관 중 “Additional personal information may be collected from users during certain NAVER service use as users access individual services, participate in

an event, or register for a giveaway event." 라는 문장을 발췌하여 간단하게 흐름을 설명할 것이다.[6] 사전에 정보 수집 및 이용의 의미를 담고 있는 collect, use 등의 단어들을 키워드로 설정해 놓았다고 가정한다.

Additional	additional	ADJ	JJ	amod	Xxxxx	True	False
personal	personal	ADJ	JJ	amod	xxxx	True	False
information	information	NOUN	NN	nsubjpass	xxxx	True	False
may	may	VERB	MD	aux	xxx	True	True
be	be	VERB	VB	auxpass	xx	True	True
collected	collect	VERB	VBN	ROOT	xxxx	True	False
from	from	ADP	IN	prep	xxxx	True	True
users	user	NOUN	NNS	pobj	xxxx	True	False
during	during	ADP	IN	prep	xxxx	True	True
certain	certain	ADJ	JJ	amod	xxxx	True	False
NAVER	NAVER	PROPN	NNP	compound	XXXX	True	False
service	service	NOUN	NN	compound	xxxx	True	False
use	use	NOUN	NN	pobj	xxx	True	False
as	as	ADP	IN	mark	xx	True	True
users	user	NOUN	NNS	nsubj	xxxx	True	False
access	access	VERB	VBP	advcl	xxxx	True	False
individual	individual	ADJ	JJ	amod	xxxx	True	False
services	service	NOUN	NNS	dobj	xxxx	True	False
,	,	PUNCT	,	punct	,	False	False
participate	participate	VERB	VB	conj	xxxx	True	False
in	in	ADP	IN	prep	xx	True	True
an	an	DET	DT	det	xx	True	True
event	event	NOUN	NN	pobj	xxxx	True	False
,	,	PUNCT	,	punct	,	False	False
or	or	CCONJ	CC	cc	xx	True	True
register	register	VERB	VB	conj	xxxx	True	False
for	for	ADP	IN	prep	xxx	True	True
a	a	DET	DT	det	x	True	True
giveaway	giveaway	ADJ	JJ	amod	xxxx	True	False
event	event	NOUN	NN	pobj	xxxx	True	False
.	.	PUNCT	.	punct	.	False	False

(그림 4) POS tagging 후 결과

(그림 4)는 tokenization, lemmatization, POS tagging을 수행한 후 결과이다. 왼쪽부터 순서대로 token, lemmatization 후 text의 원형, 대략적인 POS tag, 세부적인 POS tag, 다른 token과의 relation, token의 형태, 알파벳 여부, stop word 여부이다. 이 문장의 핵심 token인 동사, collected를 살펴보면, collected(token), collect(token의 원형), VERB(동사), VBN(past participle verb), ROOT(이 문장에서 root 역할을 수행), xxxx(대문자 없이 소문자가 연속된 형태), True(알파벳 유무 판단), False(stop word 여부 판단) 등 해당 token의 속성을 알 수 있다. 또한, collected의 원형이 사전에 명시해놓은 키워드 중 collect와 일치하기에 해당 문장으로 이후의 과정을 계속해서 진행한다.

Additional personal information	information	nsubjpass	collected
users	users	pobj	from
certain NAVER service use	use	pobj	during
users	users	nsubj	access
individual services	services	dobj	access
an event	event	pobj	in
a giveaway event	event	pobj	for

(그림 5) noun chunking 후 결과

(그림 5)는 noun chunking을 수행한 후 결과이다. 왼쪽부터 순서대로 chunk, 해당 chunk의 root token, root token과 root의 head token과의 relation, root token의 head token이다. 세 가지 token이 결합된 chunk인 additional personal information을 살펴보면, Additional personal information(chunk), information(chunk의 root token), nsubjpass(passive nominal subject), collected(chunk root token의 head token) 등 해당 chunk의 속성을 알 수 있다.

```
Additional
children: [] head: information
================================
personal
children: [] head: information
================================
information
children: [Additional, personal] head: collected
================================
may
children: [] head: collected
================================
be
children: [] head: collected
================================
collected
children: [information, may, be, from, during, access, .] head: !this is root node
================================
from
children: [users] head: collected
================================
users
children: [] head: from
================================
during
children: [use] head: collected
================================
certain
children: [] head: use
================================
NAVER
children: [] head: use
================================
service
children: [] head: use
================================
use
children: [certain, NAVER, service] head: during
================================
```

(그림 6) parse tree 중 일부분

(그림 6)은 앞선 과정을 통해 만들어낸 parse tree 중 일부분이다. 이 중 information 노드를 살펴보면, children: [Additional, personal], head: collected로 해당 노드의 children nodes와 head node를 알 수 있다.

```
users - 1.5119036458333333
use - 1.1590503472222222
service - 0.9899652777777778
services - 0.8393706597222221
event - 0.8352092013888889
information - 0.6645008680555555
```

(그림 7) textrank 후 가중치에 따른 순위

(그림 7)은 마지막으로 textrank를 실행한 후 화면이다. 가중치를 부여하는 대상은 noun인 경우로만 제한했다. 이와 같이 가중치에 따른 결과가 나오면, 해당 문장의 root token, 즉 동사의 하위 내용으로 해당 명사를 root로 가지는 chunk들이 가중치 순서대로 정렬되어 출력될 것이다.

```
[collected]
from users
access users
access individual services
during certain NAVER service use
in an event
for a giveaway event
Additional personal information
```

(그림 8) 출력되는 결과

(그림 8)은 모든 과정이 끝난 후 사용자가 보게 되는 최종 출력 화면이다. chunk의 root token, 즉 가

중치가 부여된 token의 head token이 root node가 아닌 경우에는 더욱 명료한 의미 파악을 위해 head token을 함께 출력하였다. 이를 통해 '이벤트 또는 경품 응모 서비스를 이용하는 사용자의 추가적인 개인 정보가 수집된다'는 문장의 핵심 내용을 쉽게 파악할 수 있다.

하지만, 동사가 아닌 수식어구 등으로 수집 및 이용의 의미를 담은 문장이 있다면 그 문장은 결과에 포함되지 못할 수 있다는 문제점을 인식하였다. 이를 개선하려 형용사 등의 형태까지 분석하게 되면, 오히려 사용자에게 간단한 결과를 제공하려는 기존의 취지와는 다른 결과가 도출될 수도 있다고 예상하였다. 따라서 본 논문에서는 수식어구 등까지 고려하는 것은 제외하고 실험을 진행하였다. 이러한 문제점을 해결하기 위해, 추후 연구에서는 원형이 keyword에 해당하는 동사이지만 문장에서 동사로 쓰이지 않는 token이 포함된 문장들을 따로 분류하고, 그들만의 분류 방법을 설계해 함께 출력할 수 있는 방법을 모색하여 시도할 예정이다.

4. 결론

많은 사람이 인터넷 사용에 익숙해짐에 따라, 사용자들은 셀 수 없이 많은 데이터에 노출된다. 무분별한 데이터의 공급 속에서 우리는 원치 않는 타인의 정보까지 열람하기도 하고, 그러한 일이 반복되며 자연스럽게 나 자신의 개인 정보는 안전한가? 라는 의문을 가지게 된다. 하지만, 데이터 가공을 통해 얻은 정보가 시장에서 지대한 경쟁력을 가진다는 사실을 깨달은 기업들은 어떤 방법으로든 다양한 데이터를 수집하는 데에 공을 들인다. 이러한 현실에서 소비자의 불안감은 증가하고, 자연스럽게 하락한 신뢰도는 기업에도 큰 타격이 될 수밖에 없다. 본 논문에서 약관의 핵심 키워드와 내용을 추출하는 방법을 제안함으로써, 약관에 명시되어 있는 내용을 소비자가 쉽게 인지할 수 있는 기술적 토대를 마련하고 그 결과를 구현 및 검증하였다. 이러한 기술을 토대로 약관 요약 기능을 제공하여, 소비자가 약관의 핵심 내용에 대해 인지하고 서비스를 선택하게 되어 불안은 감소하고 우려에서 기인한 기업에 대한 불신은 사라질 것이라 기대한다.

Acknowledgement

"본 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW중심대학지원사업의 연구 결과로 수행되었음"(No.2018-0-00213, SW 중심대학(건국대학교))

참고문헌

[1] Chase Buckle, Rethinking "Trust" in a New Era of Data Privacy, 2018.08.10, https://blog.globalwebindex.com/chart-of-the-week/trust-data-privacy/

[2] 서형석, 네이버 이어 카카오도…AI-이용자 대화 녹취 논란, 2019.09.03, https://www.yonhapnewstv.co.kr/news/MYH20190903021500038?did=1825m

[3] 이초희, 인터넷이용자 83.4% 개인정보 수집약관 읽지 않고 동의, 2014.10.07, https://news.naver.com/main/read.nhn?oid=277&aid=0003345980

[4] penn treebank POS tags, 2013.06.06, http://wenhoujx.blogspot.com/2013/06/penn-treebank-pos-tags.html

[5] Mihalcea, R., & Tarau, P. (2004). Textrank: Bringing order into text. In Proceedings of the 2004 conference on empirical methods in natural language processing (pp. 404-411).

[6] NAVER Privacy Policy Ver.10.2, NAVER Terms&Privacy, https://policy.naver.com/policy/privacy_en.html#a2